우리식조작체계문고

Linux봉사기 구축기술

교육성교육정보쎈터
주체97(2008)

차 례

머 리 말

위대한 령도자 **김정일**동지께서는 다음과 같이 지적하시였다.

《정보산업에 대한 인식을 바로 가지고 정보산업시대의 요구에 맞게 일하여야 합니다.》

(《**김정일**선집》 제15권, 195페지)

정보산업에 대한 인식을 바로 가지고 정보산업시대의 요구에 맞게 일하는것은 정보산업시대에 들어선 오늘 나라의 과학기술을 세계적수준으로 끌어올리며 인민경제의 정보화를 실현하는데서 매우 중요한 문제로 나선다.

정보산업시대의 도래는 콤퓨터망과 떼여놓고 생각할수 없다. 아마도 콤퓨터망이 없었더라면 콤퓨터는 지금도 연구소나 국가기관에서 사용하는 계산기계라는 정도로 사람들속에 알려져있었을것이며 정보산업의 시대는 도래하지도 않았을것이다. 콤퓨터망이 발견되고 그를 통하여 정보의 공유화가 실현되고 이러한 결과들이 생산활동에 적극적으로 도입됨으로써 콤퓨터의 지위는 계산이나 하던 도구로부터 사람들이 일상적으로 사용하는 필수품으로, 국가발전과 경제적흥망을 좌우하는 기본고리로 되였다. 이러한 요인들로 인하여 최근 세계 여러나라들에서는 콤퓨터망의 구축과 그를 통한 봉사체계를 꾸리는데 힘을 넣고 거기에 많은 관심을 돌리고있다.

우리 나라에서도 전국을 포괄하는 콤퓨터망이 구축되고 《광명》, 《내 나라》를 비롯하여 여러가지 성격의 홈페지들이 정보봉사를 진행하고있으며 많은 기관, 기업소들에서 콤퓨터를 생산과 경영활동에 적극 도입하고있다.

오늘의 현실은 최첨단기술로 무장한 정보산업시대의 주인공들을 더 많이 요구하고있다.

이 책에서는 이러한 요구를 반영하여 우리 식의 조작체계를 리용한 봉사기의 구축과 관리에서 제기되는 문제들에 중점을 두고 설명을 진행하였다.

봉사기의 구축과 관리는 콤퓨터망을 형성하고 그를 관리운영하는데서 가장 핵심적인 문제라고 할수 있다. 봉사기는 콤퓨터망의 심장이며 이러한 심장을 관리하는 봉사기관리자는 콤퓨터망의 뇌수라고도 할수 있다. 봉사기관리자의 두뇌와 능력에 의하여 콤퓨터망의 생명력이 결정된다고 할수 있다.

여기에서는 현재 가장 널리 쓰이고있는 봉사기들가운데서 관리자들이 꼭 알고 싶고 놓치지 말아야 할 봉사기의 구축과 관리방법에 대해 서술하였다. 봉사기의 관리와 운영도 다른것과 마찬가지로 더 많이 다루어보고 풍부한 경험을 쌓은 관리자들에 의하여 관리운영될 때 그 질적수준이 결정된다고 할수 있다. 우리는 나라의 정보과학기술을 세계최첨단수준으로 끌어올리겠다는 비상한 각오와 불타는 정열을 가지고 배우고 또 배워 나라의 정보산업을 세계최첨단수준에 올려세워야 한다. 우리는 이 책이 Linux조작체계를 열망하는 독자들의 좋은 방조자가 되기를 기대하면서 기탄없는 의견을 보내주기 바란다.

제1장. Linux조작체계를 리용한 콤퓨터망의 기초

콤퓨터가 세상에 자기의 첫모습을 드러낸지도 어느덧 수십년이 되였다. 이 기간에 콤퓨터는 인간생활에서 없어서는 안될 중요한 도구로, 과학연구용으로만 아니라 생활필수품으로 등장하였다.

콤퓨터는 과학, 교육, 문화, 보건, 국방 등 거의 모든 부문에서 널리 리용되고 있으며 그의 리용범위는 앞으로 더 넓어질것이다.

콤퓨터의 이러한 발전과 사용범위의 확대는 콤퓨터망의 발전과 떼여놓고 생각할수 없다. 콤퓨터망의 발전이 없었더라면 콤퓨터는 과학기술계산이나 진행하는 계산도구로서 자기의 존재를 끝마쳤을것이다.

때문에 콤퓨터망에 대해 잘 아는것은 발전하는 현실에 맞는 정보기술일군으로 준비하는데서 매우 중요한 문제의 하나로 제기된다.

무엇이나 다 그러하지만 콤퓨터망에 대해서 특히 봉사기에 대하여 배우려고 할 때 그의 기초로 되는 문제들에 대하여 잘 아는것이 매우 중요하다.

뒤에서도 언급되겠지만 봉사기는 하나의 기계이며 프로그람이라고 할수 있다. 프로그람을 동작시키는것처럼 쉬운 일이 없다는데 대해서는 많은 사람들이 잘 알고 있으리라고 본다. 그러나 프로그람을 자기가 의도하는대로 자기의 특성에 맞게 동작하도록 하는것은 매우 어려운 일이다. 그렇게 하려면 프로그람의 구체적인 동작과정은 물론 거기서 리용되는 각종 함수들과 기타 설정항목들에 대해서도 잘 알아야 하기때문이다.

이 장에서는 봉사기들의 구축과 설정에 들어가기전에 반드시 알아두어야 하는 문제들에 대해 취급한다.

이 장은 아래와 같은 체계로 되여있다.

- Linux 콤퓨터망의 설치를 위한 준비작업
- Linux조작체계에서 사용되는 지령들
- Linux조작체계의 등록부구조

제1절. Linux콤퓨터망의 설치를 위한 준비작업

Linux는 1991년 핀란드 헬싱키대학의 학생이였던 리누스 토발즈(Linus B. Tovalds)에 의해 개발된 조작체계이다. Linux는 그 당시 intel386기종에서 동작하던 XENIX라는 조작체계를 모형으로 하여 만들어졌으며 개발코드를 인터네트상에 공개하면서 많은 사람들이 관심을 가지게 되였다. 그후 전 세계의 개발자들이 다같이 Linux개발에 참가하여 현재는 매우 뛰여난 성능을 자랑하는 조작체계로 되였다.

Linux의 최대우점은 뛰여난 콤퓨터망기능이다. Linux핵심부자체가 다양한 콤퓨터망기능을 지원하며 웨브봉사기와 우편봉사기, 령역이름(DNS)봉사기, 파일봉사기 등 다양한 봉사프로그람과 합쳐져서 강력한 콤퓨터망기능을 제공한다.

1.1.1. Linux에서 콤퓨터망을 사용하기 위한 준비과정

Linux에서 콤퓨터망봉사를 사용하기 위해서는 다음과 같은 과정을 거쳐야 한다.

- 망대면부기판의 설치

망대면부기판을 설치한다. 망(LAN)기판이나 모뎀, 케블 등이 여기에 해당된다. 콤퓨터에 망기판이나 모뎀을 설치하고 거기에 케블을 련결하면 물리적인 련결과정은 끝난다고 볼수 있다.

- 망규약의 설치

봉사프로그람이 사용하는 규약을 설치한다. 규약에는 인터네트에서 사용되는 TCP/IP나 SLIP, PPP 등이 있다. Linux는 이런 규약들을 기본적으로 제공한다.

- 망봉사프로그람의 설치

실지로 사용할 콤퓨터망봉사프로그람을 설치한다. 웨브봉사를 위한 웨브봉사기나 우편봉사기 등 여러가지 봉사기프로그람 등이 여기에 해당된다.

그러면 이러한 준비과정에 대해 하나씩 보도록 하자.

1.1.2. 콤퓨터망대면부 기판의 설치

Linux에서 콤퓨터망을 사용하기 위해서는 먼저 콤퓨터망대면부기판를 설치하고 Linux핵심부에 인식시켜야 한다. Linux에 설치할수 있는 종류와 설치하는 방법은 다음과 같다.

- Linux에서 지원하는 콤퓨터망대면부

Linux는 다양한 콤퓨터망의 하드웨어를 지원하며 이들의 대면부를 항목별로 분류해 놓았다. 그리고 같은 종류의 대면부를 사용할 때는 번호를 정하여 eth0, eth1 로 표기하여 구분한다. 매개 종류는 다음의 표와 같다.

표 1-1. 망대면부의 종류

대면부의 종류	설 명
lo	local loop back대면부를 가리킨다. 이것은 시험을 목적으로 사용하는 대면부로서 망기판이 없더라도 제공된다. lo의 IP주소는 127.0.0.1이며 이것을 통해서 콤퓨터망의 시험을 할수 있다.
eth*n*	일반적으로 사용하는 망기판인 Ethernet대면부를 가리킨다. 망기판은 여러장을 동시에 사용할수 있으며 개개는 번호로서 구분한다. 즉 eth0, eth1이런 식으로 구분한다.
dl*n*	이것은 D-Link DE-600형의 대면부이다. 이것은 병렬포구를 통해 구동하는 DE-600에서만 사용된다.
sl*n*	이것은 SLIP대면부를 가리킨다.
ppp*n*	이것은 PPP대면부를 가리킨다. 가정에서 사용하는 ADSL모뎀으로 콤퓨터망을 사용할 때에는 대부분 ppp대면부를 사용한다.
plip*n*	이것은 병렬케블을 사용하는 병렬대면부를 가리킨다.

물론 이외에도 ISDN이나 AX.25, ATM 등 다양한 콤퓨터망하드웨어를 지원한다.

- 망기판 설치방법

망기판을 설치하는 방법은 매우 간단하다. Linux Kernel 2.4이상에서는 설치과정에서 자동으로 망기판을 인식한다. 망기판에 따라서 특성이 다르기는 하지만 PCI방식의 망기판을 사용한다면 대부분의 핵심부가 자동적으로 인식한다.

이전에 나온 ISA망기판을 사용하는 경우에는 자동으로 인식이 될수도 있고 안될수도 있다. 기판인식이 잘 안되는 경우에는 IO주소와 IRQ번호를 알고 있으면 수동으로 기판을 인식시킬수 있다. 수동으로 기판을 설치할 때에는 아래와 같이 모듈을 추가하면 된다.

- 콤퓨터망의 모듈 선택하기

Linux는 필수적인 몇가지 기능을 제외하고는 다른 나머지 기능들은 모듈화시켜서 필요할 때마다 쓸수 있게 해두었다.

모듈(module)이란 무엇인가?

모듈은 Linux 핵심부의 핵심적인 기능의 하나이다. 이것은 핵심부의 모든 기능을 통채로 가지고있는것이 아니라 여러개의 부분으로 갈라서 거기서 필요한 부분만을 바꿔가면서 사용하는 방법이다. 모듈을 사용하면 핵심부의 크기를 최적화할수 있으며 모듈을 바꿀 때마다 콤퓨터를 재기동하지 않아도 되는 등 여러가지 우점이 있다.

그림 1-1. Linux핵심부과 모듈사이 관계

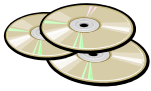
콤퓨터망 구동프로그람도 모듈형태로 제공되기때문에 망기판에 필요한 모듈을 그때마다 설치하면 된다. 콤퓨터망 구동프로그람 모듈은 /lib/modules/<kernel 판본>/net 등록부에서 찾을수 있다. Kernel 판본 2.6.6 에서 지원하는 구동프로그람의 모듈에 대해서는 아래와 같이 알아볼수 있다.

그림 1-2. Linux에서 지원하는 콤퓨터망 하드웨어

가장 많이 사용하는 NE2000 PCI판본은 ne2k-pci.o이다. 3Com계렬의 망기판은 3c503.o, 3ct509, 3c59x 등이며 Intel의 ether express계렬은 eepro100.o eexpress.o 이다.

그 외에도 모뎀사용자를 위한 ppp.o나 slip.o 등의 모듈도 있다.

설치하려는 콤퓨터망기판의 모듈이 어떤것인지 모르는 독자들은 /boot/modules-info

파일을 참조하여야 한다. 이 파일에는 모듈의 이름과 실제 상표이름이 나와있다. 아니면 핵심부원천등록부인 /usr/src/linux/Documentation/networking/net-modules. txt파일을 참조해도 된다. Ethernet기판에 대한 자세한 정보를 원한다면 /usr/doc/HOWTO/Ethernet-HOWTO를 참조하면 된다.

실례 1-1: modules-info파일의 일부

```
3c501
        eth
         "3Com 3c501 (throw it away!)"
        io  "Base I/O address"   "0x[0-9a-fA-F]+"
        irq  "IRQ level"  "[0-9]+"

3c503
        eth
         "3Com EtherLink II"
        io  "Base I/O address"  "0x[0-9a-fA-F]+"
        irq  "IRQ level"  "[0-9]+"
        xcvr  "Transceiver (0 = BNC; 1 = AUI)"  "[01]"

eepro
        eth
 "EtherExpress Pro/10"
        io  "Base I/O address"  "0x[0-9a-fA-F]+"
        irq  "IRQ level"  "[0-9]+"
```

- 모듈의 설치

모듈을 핵심부에 추가시킬 때는 modprobe지령을 사용한다. 사용방법은 다음과 같다.

```
modprobe <모듈이름> <추가선택항목>
```

이것을 사용한 실례는 아래와 같다.

```
# modprobe eepro
혹은
# modprobe ne.o io=0*300 irq=5
```

모듈이름을 적을 때는《.o》를 적어도 되고 적지 않아도 된다. PCI기판의 경우는 IO와

IRQ를 직접 적어야 한다. 대부분의 ISA기판은 DOS에서 IO와 IRQ를 직접 적어야 한다. ISA기판은 DOS에서 IO와 IRQ를 설정하는 프로그람이 있기때문에 이것을 사용해서 IO와 IRQ값을 조정해주고 Linux로 다시 들어와서 modprobe를 사용하면 된다. 모듈을 핵심부에 성공적으로 설치하였다면 IRQ와 IO를 출력하여 본다. 모듈을 인식시키면 그때부터 망기판을 사용할수 있다.

- 콤퓨터망모듈의 확인과 삭제

설치된 모듈을 확인할 때에는 lsmod를 사용한다. lsmod를 사용하면 현재 설치된 모듈의 이름과 그에 관한 설명을 볼수 있다. 그림 1-3에 그에 대한 실례를 주었다. 설치된 모듈을 삭제할 때에는 rmmod지령을 사용한다.

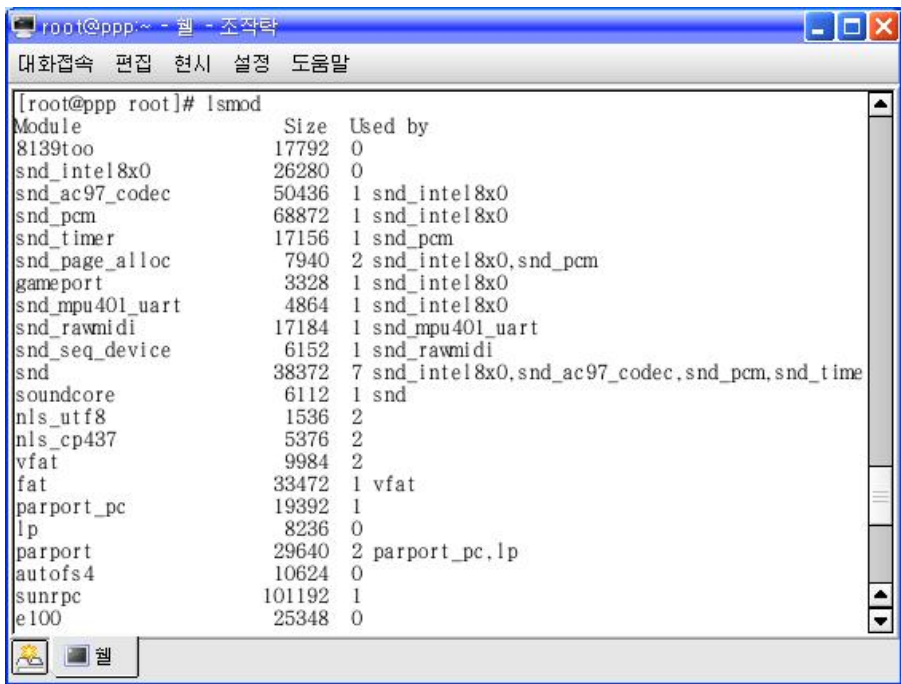



그림 1-3. lsmod의 사용실례

- 모듈설치를 자동으로 진행하기

모듈을 몇번 설치해보느라면 콤퓨터를 켤 때마다 자동으로 모듈을 설치할수는 없을가 하는 생각이 떠오르게 된다. 물론 방법은 있다. Linux는 핵심부모듈의 자동설치를 위해서 kerneld를 지원한다. 이때는 /etc/modules란 파일에 설치할 모듈을 다음과 같이 적어주면 된다.

```
alias eth0        ne
options ne        io=0*300 irq=5
```

첫번째줄은 eth0에 NE2000인 ne.o모듈을 설치하라는것이고 두번째줄은 설치추가선택항목으로 io=0×300번과 irq=5번으로 하라는것이다.

- 핵심부안에 넣기

모듈을 설치하기가 싫으면 아예 핵심부화상자료안에 집어넣을수도 있다. 사실 이 방법은 Linux가 모듈기능을 지원하기전에 사용하던 방법이다. 이를 위해서는 핵심부콤파일선택항목에서 콤퓨터망기판을 선택하고 콤파일해서 사용하면 된다. 핵심부의 콤파일후에는 /etc/lilo.conf파일에 다음과 같이 망기판의 설정내용을 적어넣는다.

```
append= “ether=5, 0*300, eth0”
```

append는 장치를 추가할 때 사용하는 지령이다. ether는 Ethernet을 가리킨다. 그리고 IRQ=5, IO=0*300을 eth0에 할당한다는 뜻이다. 그 다음 lilo.conf파일을 보관하고 /sbin/lilo를 실행하면 변경된 내용을 Linux핵심부화상파일에 등록시킨다. 그후에 체계를 재기동하면 망기판을 인식한다.

1.1.3. Linux 핵심부 콤파일

Linux에서 핵심부의 역할과 핵심부를 콤파일하는 방법을 알아보자. 여기에서는 Linux kernel 2.6을 기준으로 설명한다.

- Linux 핵심부란?

우리가 핵심부라고 하는 Kernel의 사전적인 의미는 식물의 씨나 어떤 문제의 핵심을 나타낸다. 조작체계에서 핵심부는 체계의 자원을 관리하는 가장 핵심적인 부분이다. 핵심부는 기억기에 상주하여 체계장치와 기억기프로쎄스를 관리하고 입출력을 처리한다. 사실상 Linux는 엄밀히 말하면 Linux핵심부를 의미하는것이다. 다시말하여 Linux는 Linux핵심부와 응용프로그람으로 구성되여있다. Linux체계를 사용하는데서 다음과 같은 문제들이 제기된다.

- 체계의 최적화

Linux배포판에 포함되여있는 기본 핵심부에는 거의 사용되지 않는 다양한 기능들이 들어있어 실제로 필요한 핵심부보다 크기도 크고 성능이 떨어질수도 있다. 그래서 많은 Linux사용자들은 자기의 체계에 적합한 작은 핵심부를 새로 만들어 Linux 체계의 성능을 최적화한다.

- 새로운 기능의 사용

핵심부를 콤파일하는 또 다른 리유는 새 핵심부를 만들어서 향상된 기능을 사용하거나 새로운 장치를 지원하기 위해서이다. Linux 핵심부는 매우 빠르게 판본이 올라간다. 보통 한달에 한번씩 판본이 올라가기 때문에 판본이 올라갈때마다 추가되는 새로운 기능을 사용하기 위해서 핵심부를 콤파일하는 Linux사용자들도 많다.

- Linux 핵심부 판본

Linux 핵심부는 매우 빨리 개발되고 배포된다. 핵심부의 개발자들은 안전성과 발전을 위해 Linux핵심부를 두가지 판본으로 분류해놓았다.

- 개발판본과 안정판본

새로운 기능들이 실험되는 개발과정의 핵심부를 개발판본이라고 부른다. 이것은 주로 새로운 기능을 원하는 개발자나 시험을 위한 판본이다. 개발판본은 많은 시험과 보수(patch)과정을 거쳐 안정판본으로 배포된다. 따라서 일반사용자들은 개발판본보다는 안정판본을 사용하는것이 좋다.

- 핵심부의 판본번호

핵심부의 판본번호는 2.4.20또는 2.5.10과 같이 세부분으로 구성되여있다. 처음의 수자는 커다란 변화가 있을 때 바뀌는 주판본의 번호이다. 이것은 핵심부에 중대한 변화가 있을 때 적용한다. 그리고 가운데 수자는 핵심부의 종류를 나타낸다. 짝수일 때 안정판본을 의미하고 홀수일 때는 개발판본을 의미한다. 즉 2.5.X의 개발판본이 안정화되면 2.6.X의 안정판본으로 배포된다. 마지막 수자는 보수준위를 나타낸다.

- 핵심부 콤파일도구

핵심부를 콤파일하기 위해서는 C콤파일러와 make를 비롯한 몇가지 프로그람이 필요하다. 이 프로그람들은 Linux 배포판 CD에 있기 때문에 먼저 설치여부를 확인하고 필요한것은 추가로 설치한다.

표 1-2. 핵심부콤파일도구

도 구	최 소 판 본	확 인 방 법
GNU C Compiler	2.91.66	#gcc-version
CNU Make	3.77	#make-version
Binutils	2.9.1.0.25	#ld-v
util-linux	2.10	#fdformat-version
Modutils	2.4.2	#insmod-V
e2fsprogs	1.19	#tune2fs
Reiserfsprogs	3.x.0b	#reiserfsck 2>&1|grep reiserfsprogs 이것은 Reiser파일 체계를 사용할 때만 필요하다.
pcmcia-cs	3.1.21	#cardmgr-V 이것은 PCMCIA를 사용할 때만 필요하다.
PPP	2.4.0	#pppd-version
isdn4k-utils	3.1pre1	#isdnctrl 2>&1|grep version 이것은 ISDN을 사용할 때만 필요하다.

• 핵심부콤파일의 시작

그러면 실지로 핵심부의 콤파일을 진행해보자.

핵심부의 콤파일은 다음의 과정을 거쳐 진행된다.

- 핵심부원천코드의 설치

먼저 root권한을 가지도록 한다. 그리고 /usr/src/linux등록부에 핵심부원천코드가 있는지 확인한다. 보통 그안에 header등록부만 있으면 핵심부의 머리부만 있는것이다. 이러한 경우에는 핵심부의 원천rpm파일을 구해서 설치한다. tar.gz형태의 핵심부원천코드를 설치하는 방법은 /usr/src등록부에 들어가 기존의 원천이 있으면 삭제하고 핵심부등록부를 만든 다음에 tar지령을 사용해서 핵심부 원천코드를 풀면 된다.

```
# cd /usr/src
# rm -f linux
# mkdir linux-2.4.X.tar.gz linux-2.4.X
# ln -s linux-2.4.X linux
```

- 핵심부원천코드의 보수

Linux핵심부는 매우 빠른 속도로 보수(patch)된다. 보통 일주일에 한번씩은 보수된 핵심부원천코드가 공개되기때문에 핵심부가 보수될 때마다 전체 원천코드를 다시 내리적재하는것은 비효률적이다. 이때는 핵심부보수파일만 받아서 핵심부원천코드를 보수하면 된다.

핵심부의 보수는 diff소응용프로그람을 리용하여 핵심부의 일부분을 수정하는 방법으로 진행된다. 보수파일은 빠르게 갱신되며 새로운 보수파일은 이전 보수에 대한 내용은 포함하지 않는다. 따라서 보수파일을 사용할 때는 보수파일을 순서대로 모두 적용해야 한다. 보수파일과 원천코드파일의 판본간격이 너무 길면 보수를 하는것보다 원천코드를 다시 콤파일하는것이 효과적이다.

Kernel 2.4.13에서 Kernel 2.4.16로 갱신하는것을 실례로 들면 patch-2.4.14.gz, patch-2.4.15.gz, patch-2.4.16.gz를 순서대로 적용해야 한다.

아래에 그의 적용과정을 서술하였다.

```
# cd /usr/src
# ls
linux linux-2.4.13 patch-2.4.14.gz patch-2.4.15.gz patc
h-2.4.16.gz
# gzip - cd patch-2.4.14.gz | patch -p0
# gzip - cd patch-2.4.15.gz | patch -p0
# gzip - cd patch-2.4.16.gz | patch -p0
```

추가선택항목가운데 p0은 보수파일에 들어있는 경로에 대한 상대적인 위치를 나타낸다. 보수에 성공했다면 보수대상이 된 파일의 원본은 이름 끝에 《.orig》를 붙여서 여벌복사된다.

보수과정에서 문제가 발생하여 실패했다면 실패한 파일이름 뒤에 《.rej》를 붙인 파일을 만든다. 이때는 *.rej파일을 살펴보고 다시 보수한다.

- make mrproper

이전에 설정한 정보를 지우거나 새로운 핵심부원천코드를 가져다 콤파일할 때에는 make mrproper지령을 사용한다.

```
# make mrproper
```

이 지령은 이전의 핵심부를 콤파일할 때 만들어진 object파일(*.o)과 의존성설정, 콤파일의 환경설정값, 판본정보 등 새로 시작하는 콤파일에 영향을 주는 이전 정보들을 삭제한다. 이전에 설정된 콤파일환경을 재사용한다면 이 과정을 건너뛰여도 된다.

- 콤파일 환경설정

Makefile을 편집해서 핵심부가 설치될 등록부를 결정한다.

```
INSTALL_PATH=/boot (맨 앞부분의 주석(#)을 제거한다.)
```

핵심부의 콤파일을 진행하려면 현재 사용자의 하드웨어의 수준에 맞도록 핵심부에 사용되는 여러가지 추가선택항목을 설정해야 한다. 핵심부콤파일환경설정을 위한 대면부로는 make config, make menuconfig, make xconfig 세가지가 있다.

- make config

이것은 가장 기본적인 설정방법이다. 사용자의 하드웨어의 구성을 순서대로 물어본다. 하지만 중간에 설정을 잘못하면 처음부터 다시 해야 하기때문에 사용하지 않는것이 좋다.

- make menuconfig

하드웨어항목들을 TEXT환경에서 차림표로 보여준다. 건반을 사용해서 선택만 하면 되기때문에 쉽게 환경을 설정할수 있다. 또한 X-Window를 사용하지 않고 telnet를 사용해서 원격지에서도 편리하게 사용할수 있는 우점이 있다.

차림표방식의 화면에서 지름건과 방향건으로 차림표사이를 이동할수 있다.

설정이 끝났으면 기본차림표에서 “Save Configration to an Alternate File”을 선택하여 설정내용을 파일로 보관한다. 보관된 파일을 다시 읽어들일 때는 “Load an Alternate Configuration File”차림표를 선택한다. make menuconfig는 ncurses(new-curses) Library를 사용하기때문에 아직 ncurses가 설치되지 않았다면 먼저 이것을 설치한다.

- make xconfig

X환경에서 차림표를 보여준다. 사용하는 방법은 menuconfig와 거의 같다. X-Window를 사용하고 있으면 이 방법도 사용해 볼 가치가 있다. Make xconfig를 수행하기 위해서는 X-Window와 Tcl/TK해석기/도구합서고가 필요하다.

설정한 내용은 기본화면의 "Store Configuration to File"항목에서 파일로 보관할수 있다. 콤파일환경은 /usr/src/linux/arch/i386/config.in에 보관된다. <M>으로 표시되는것은 핵심부안에 접어넣지 않고 모듈로 콤파일해서 따로 모아둔다는 뜻이다. 핵심부설정을 마쳤으면 설정한 내용을 보관하고 탈퇴한다.

- 핵심부콤파일

- 핵심부등록부의 정돈

핵심부의 콤파일환경설정이 모두 끝났으면 콤파일을 시작하기전에 사용한 object파일을 지운다.

- make dep: 콤파일을 위한 의존성관계를 설정한다.
- make clean: 이전에 수행했던 콤파일과정에서 생성된 object파일(*.[oas]), 핵심부, 림시파일(*.tmp), 설정값(configure) 등을 삭제한다.

```
# cd /usr/src/linux
# make dep
# make clean
```

- 핵심부 콤파일

핵심부를 콤파일한다. 핵심부의 콤파일은 make zImage를 사용한다. 이 과정에서 시간이 가장 많이 걸린다. 핵심부의 콤파일이 성공하면 /usr/src/linux/arch/i386/boot등록부에서 핵심부화상파일이 생성된다.

```
# make zImage
```

핵심부를 콤파일했을 때 제한된 크기를 넘으면 "System is too big"라는 오유통보문을 남기고 콤파일이 실패한다. 이때는 잘 쓰지 않는 기능은 모듈로 설정하고 필요없는 기능은 제거하여 핵심부의 규모를 줄이고 다시 콤파일한다. 또는 make bzImage를 사용하면 큰 화상파일을 만들수 있다. 핵심부의 화상파일이 만들어졌으면 make install을 사용해서 만들어진 핵심부화상파일을 설치한다.

```
# make install
```

- 핵심부모듈의 콤파일

핵심부의 설정과정에서 <M>으로 설정한것들은 make modules추가선택항목을 사용하여 콤파일을 진행한다.

```
# make modules
```

모듈을 콤파일했으면 make modules_install을 사용해서 설치한다.

```
# make modules_install
```

- 핵심부의 설치

핵심부화상파일을 만들었으면 이것을 boot등록부로 옮기고 lilo지령을 실행시킨다. 먼저 만들어진 기동화상파일을 /boot등록부로 복사한다.

```
# cp /usr/src/linux/arch/i386/bzImage /boot/zImage-2.4.16
```

그리고 /etc/lilo.conf파일의 핵심부화상등록부를 새로운 핵심부화상파일이 있는 등록부로 변경한다. 보통 lilo.conf에는 사용용도에 따라서 여러개의 화상파일이 설정되여있다. 새로 생성한 화상파일이 zImage-2.4.16이기때문에 이것을 image에 추가시키고 label에 이것을 표시한다. Label은 체계기동시 화면에 표시되는 이름이다.

```
boot=/dev/had
image=/boot/vmlinuz
label=linux
root=/dev/hdb1

image=/boot/zImage-2.4.16
label=new
root=/dev/hdbl
other=/dev/hda1
```

마지막으로 /sbin/lilo를 실행해서 핵심부화상파일을 갱신한다. 이제 체계를 재기동하면 새로운 핵심부로 기동하는것을 볼수 있다.

1.1.4. Kernel 2.6에서 콤퓨터망 추가선택항목

Kernel 2.6.X는 강력한 콤퓨터망기능을 제공한다. 여기에서는 Kernel 2.6을 콤파일할 때 설정하는 콤퓨터망의 추가선택항목에 대해서 보도록 하자.

- Packet Socket

파케트규약은 tcpdump프로그람처럼 socket와 같은 콤퓨터망계층을 사용하지 않고 핵심부준위에서 콤퓨터망장치와 직접 통신하는 방법을 제공한다. 이것은 모듈로 사용할수 있으며 이때는 af_packet.o로 만들어진다.

- Kernel/User network link driver

이것은 핵심부와 사용자프로쎄스사이에 쌍방향통신을 허락한다. 사용자프로쎄스는 /dev등록부에서 해당 내용을 읽고 쓸수 있다. arp데몬을 사용하기 위해서는 이것을 ‘Y’로 설정해야 한다.

- Network packet filtering

파케트려과를 위하여 사용한다. 이것은 이전 판본의 ip masquerading기능을 대신하는 장치이다. Kernel 2.2에서는 ipchains를 사용했지만 Kernel 2.4부터는 iptables지령을 사용한다.

- Unix Domain Socket

Unix Domain Socket는 소케트방식의 IPC(Inter Process Communication)이다. 이것을 사용하면 소케트를 생성해서 내부프로세스사이에 정보를 교환할수 있다.

- TCP/IP networking

TCP/IP콤퓨터망 기능이다. 인터네트를 사용하기 위해서는 이것을 반드시 선택해야 한다. 필요한 항목들을 보기로 하자.

- IP multicasting: IP복수수신자송신기능이다. 프로그람중에 복수수신자를 사용하는것이 있다면 이것을 활성시켜야 한다.
- IP Advanced Router: Linux 체계를 경로기로 사용할 때 설정한다. 이 추가선택항목은 일반적으로 경로기기능에 려과, 보안(례를 들어 IP속이기 예방) 등 몇가지 향상된 기능을 지원한다.
- IP Tunneling: VPN과 같이 물리적으로 멀리 떨어진 콤퓨터망에서 한 규약안에 다른 규약의 자료를 넣어 전송할 때 사용한다.
- IP TCP syn cookie support: 콤퓨터망상에서 DoS(Denial of Service) 등 TCP의 SYN공격으로부터 체계를 보호해 준다. 하지만 체계에 부하가 많이 걸린다면 이것을 사용하지 않는것이 좋다.
- IPv6 protocol: IPv6을 지원한다. IPv6을 사용하는 콤퓨터망에 있다면 이것을 활성화시킨다.

- ATM(Asynchronous Transfer Mode)

Linux 체계에서 IPX를 사용해서 Novell콤퓨터망를 사용할 때 설정한다.

1.1.5. 콤퓨터망 대면부 기판 설정

핵심부가 망기판을 인식한후에는 망기판에 고유한 주소를 할당하고 동작시키는 과정이 필요하다. 콤퓨터망대면부를 설정할 때에는 ifconfig(interface config)지령을 사용한다. ifconfig지령을 사용하면 다음과 같은 일을 할수 있다.

- 설치된 콤퓨터망 대면부의 확인

기본적으로 ifconfig지령을 사용하면 현재 설치된 콤퓨터망대면부를 보여준다. 콤퓨터망장치가 설치되지 않았다면 다음과 같은 결과를 보여준다.

그림 1-4. ifconfig의 실행(1)

첫행의 lo라는것은 loopback only의 략자이다. 그 다음행의 내용은 아래와 같다.

표 1-3. ifconfig에서 사용되는 항목

항 목	설 명
Inet addr	대면부의 IP주소를 의미한다.
Mask	대면부의 NetMask를 의미한다.
Bcast	대면부의 broadcast주소를 의미한다.
MTU	Maximum Transfer Unit의 략자로서 대면부가 한번에 전송할수 있는 파케트의 크기이다.
RX, TX	송신하고 수신한 자료에 관한 정보를 보여준다.

- 콤퓨터망 대면부의 확인

ifconfig의 추가선택항목으로 콤퓨터망 대면부 이름을 적어주면 해당 대면부에 대한 정보를 보여준다.

```
ifconfig [대면부 이름]
```

Ethernet대면부에 대한 정보를 보고싶을 때에는 ifconfig지령에 해당 대면부를 지정하면 된다. 그림 1-5는 Ethernet 기판인 eth0에 대한 정보를 보여준것이다.

Ethernet대면부는 lo와는 다른 고유한 하드웨어 주소를 가지고 있는데 Hwaddr항목에서 주소값을 볼수 있다. 또한 고유한 IP주소를 가지고있다.

그림 1-5. ifconfig의 실행(2)

- 대면부의 기동

ifconfig를 사용하면 콤퓨터망의 대면부를 기동시킬수 있다. 대면부가 TCP/IP를 사용할 때에는 IP주소도 같이 설정해줄수 있다.

```
ifconfig <대면부> <IP 주소> [추가선택항목] up
```

우와 같이 대면부의 이름을 적고 할당받은 IP주소를 입력한다. 그리고 대면부의 시작을 알리는 up을 기입한다. 하지만 이것은 반드시 핵심부에 eth0을 설치한 다음에 해야 한다.

정상적으로 설정이 된 경우에는 아무런 통보문도 발생하지 않는다. 보통 IP 주소를 적으면 해당되는 netmask 를 자동으로 설정해준다. 하지만 A,B,C 클라스가 아닌 다른 부분콤퓨터망을 사용할 때에는 netmask 도 적어야 한다. 가령 C 클라스를 네개의 부분콤퓨터망으로 쪼개면 netmask 255.255.255.64 로 설정하여 주어야 한다.

- 대면부기동중단

ifconfig는 콤퓨터망대면부를 시작하는 기능을 수행하는것과 함께 중단하는 지령으로도 사용할수 있다. 이때는 대면부이름을 적고 down지령을 사용하면 된다.

```
ifconfig <대면부 이름> down
```

주의할것은 이것이 단지 대면부의 동작을 중단시킨다는것뿐이다. 핵심부는 하드웨어적인 콤퓨터망대면부를 여전히 인식하고있는 상태이다.

즉 대면부는 up/down지령을 통하여 언제든지 기동시키거나 중단시킬수 있다. 설정한 대면부의 결과는 그림 1-6과 같다.

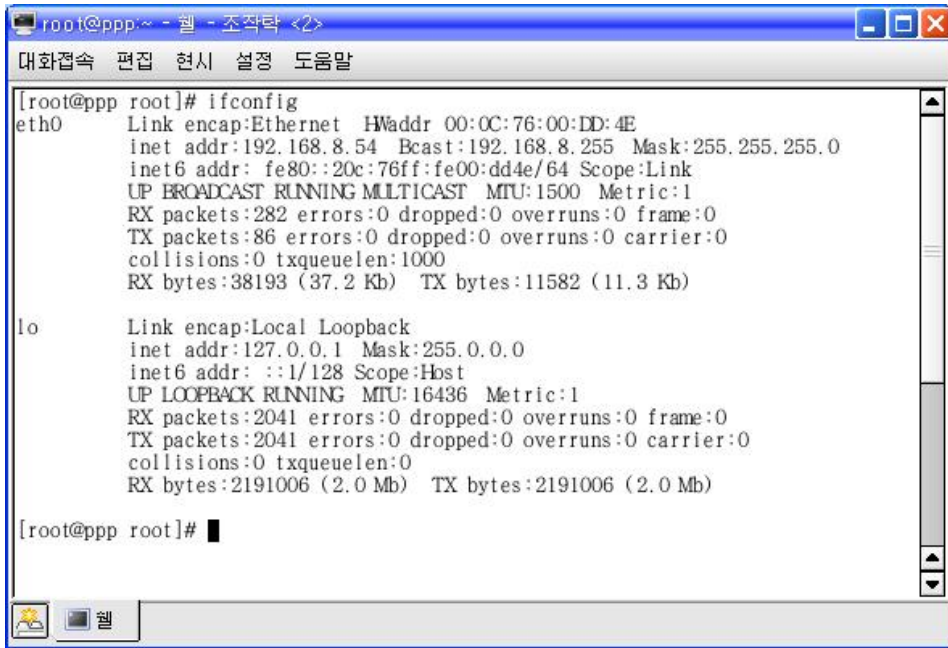



그림 1-6. ifconfig의 실행(3)

- 콤퓨터망스크립트 파일의 설정

지금까지 ifconfig지령을 사용하여 콤퓨터망 대면부기판을 설정하는 방법을 보았다. 하지만 이 지령은 Linux로 콤퓨터를 기동할 때마다 매번 사용해야 한다는 약점이 있다.

그러나 콤퓨터망스크립트 파일에 ip정보를 보관해두면 콤퓨터가 기동할 때 자동으로 이것을 설정한다. 스크립트파일은 /etc/sysconfig/network-scripts등록부에 있으며 파일의 이름은 ifcfg-'*' (*: 대면부 이름)으로 되여있다. 즉 eth0, eth1 망대면부는 ifcfg-eth0, ifcfg-eth1이다. 스크립트 파일에 설정하는 내용은 다음과 같다.

표 1-4. 스크립트파일의 설정항목

항 목	설 명
DEVICE	설정할 장치이름을 입력한다.
BOOTPROTO	기동시 사용할 규약을 입력한다. DHCP로 동적 IP를 사용하면 'dhcp'를 입력하고 정적 IP방식을 사용하면 'static'를 입력한다.
BROAKCAST	대면부의 방송주소를 입력한다.
IPADDR	대면부의 IP주소를 입력한다.
NETMASK	대면부의 NETMASK주소를 입력한다.
NETWORK	대면부의 콤퓨터망 주소를 입력한다.
ONBOOT	기동시 이 대면부를 사용할것인가를 설정한다. 사용하면 'yes'를 입력한다.

- DCHP 주소설정의 실례

DCHP를 사용하면 DEVICE, BOOTPROTO, ONBOOT항목만 입력하면 된다. eth1대면부를 DCHP로 설정한 실례는 다음과 같다.

```
DEVICE=eth1
BOOTPROTO=dhcp
ONBOOT=yes
```

- 정적 IP주소설정의 실례

정적 IP를 사용하면 DEVICE, BOOTPROTO, BROADCAST, IPADDR, NETMASK, NETWORK, ONBOOT 항목을 모두 설정한다. C클라스의 IP주소인 《192.168.8.54》를 설정한 경우에는 그림 1-7과 같이 나온다.

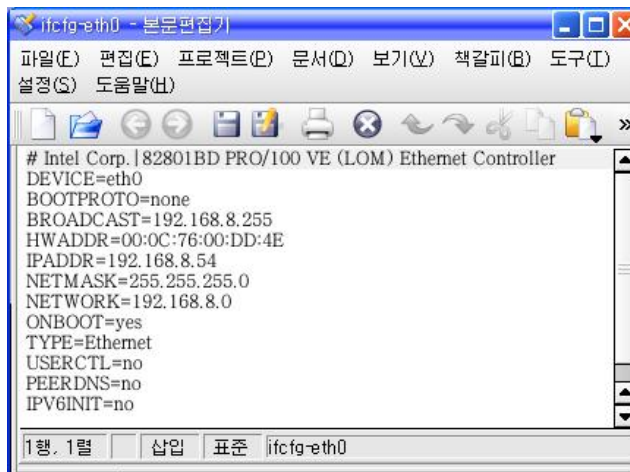



그림 1-7. 콤퓨터망의 스크립트파일

제2절. Linux조작체계에서 사용되는 지령들

앞에서 우리는 Linux조작체계에서 콤퓨터망을 설치하기 위한 필요한 준비과정에 대하여 간단히 보았다. 여기서는 Linux조작체계에서 사용되는 기본지령들에 대하여 간단히 취급한다. 조작체계의 사용에서 지령들이 얼마나 중요한 역할을 하는가에 대해서는 콤퓨터를 다루어본 사람이라면 누구나 쉽게 리해하리라고 본다.

그러면 Linux조작체계에서 사용되는 기본 지령들에 대해 보도록 하자.

cd

cd지령은 'Change Directory'의 략자로서 작업등록부를 이동할 때 사용하는 지령이다. 문법은 아래와 같다.

```
$cd 〔등록부이름〕
```

등록부이름을 표기할 때에는 등록부의 절대경로와 상대경로를 모두 사용할수 있다.

만약 사용자의 홈등록부에 있는 《bin》이라는 등록부로 작업등록부를 변경할 경우 아래와 같이 변경할수 있다.

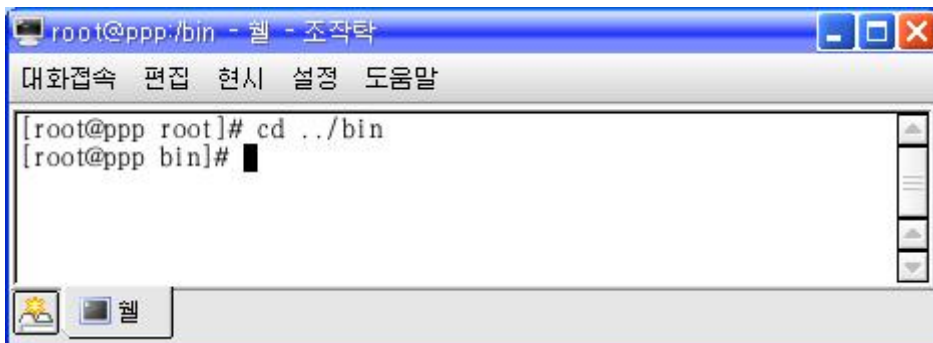



그림 1-8. 상대경로를 리용한 등록부의 변경

그림 1-8은 상대경로를 사용하여 《bin》등록부를 작업등록부로 변경하기 위해 상위등록부를 의미하는 《..》을 사용하였다.

이와는 달리 최상위등록부에서 《bin》등록부의 절대경로를 설정해서 등록부를 변경할수도 있다.

ls

작업등록부의 파일이나 등록부들의 목록을 출력해주는 지령이다. 문법은 아래와 같다.

$ ls 〔OPTION〕 〔FILE/DIRECTORY〕

그림 1-9와 같이 지령을 주면 등록부에 있는 파일이나 등록부의 목록을 볼수 있다.

그림에서 볼수 있는바와 같이 root등록부안에 보관되여 있는 각종 등록부와 파일 등의 목록들을 출력한다.

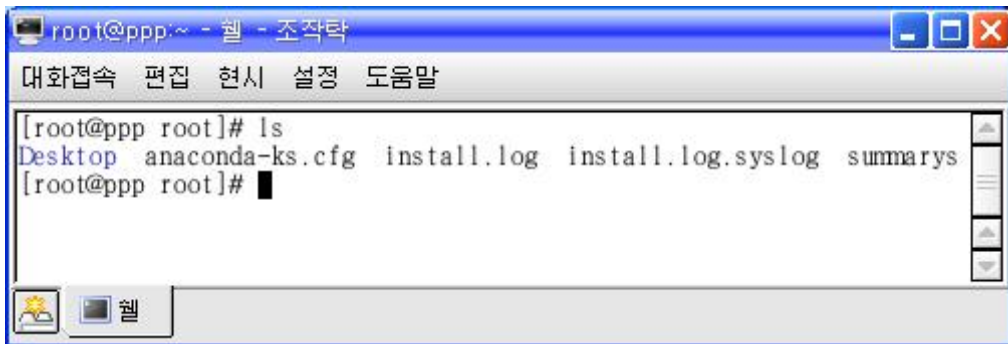



그림 1-9. ls의 실행화면

ls지령은 또한 여러가지 추가선택항목을 제공한다. 표 1-5는 ls 지령에 추가할수 있는 추가선택항목들이다.

표 1-5. ls지령의 추가선택항목

추가설정항목	설 명
-a	등록부의 모든 파일과 등록부목록을 출력한다.
-A	-a와 기능은 같지만 《.》, 《..》을 출력하지 않는다.
-b	출력할수 없는 문자는 8진수로 출력한다.
-B	~으로 끝나는 목록을 출력하지 않는다.
-c	파일의 생성시간을 기준으로 목록을 출력한다. -I추가선택항목과 함께 사용할 경우 파일의 생성시간을 출력한다.
-C	10개씩 목록을 출력한다.
-d	등록부의 파일대신 등록부를 출력한다.
-D	Emacs 에 맞게 목록을 출력한다.
-i	파일의 inode와 함께 목록을 출력한다.
-I	파일의 권한, 소유자, 그룹, 크기, 날자 등과 함께 출력한다.
-m	파일을 《.》로 구분하여 가로방향으로 목록을 출력한다.
-r	목록출력에 대한 정렬추가선택항목이 선택되였다면 정렬추가선택항목에

	대해 거꾸로 목록을 출력한다.
-s	파일 크기가 kb단위인 목록을 출력한다.
-t	파일의 생성날자가 가장 늦은 차례로 목록을 출력한다.
-x	목록을 세로방향으로 출력한다.
-f	목록을 정렬하지 않는다.
-F	파일의 형태와 함께 목록을 출력한다.
-R	모든 부분목록을 포함해서 목록을 출력한다.
-S	파일의 크기순서로 목록을 출력한다.
-U	정렬하여 목록을 출력한다.
-1	출력행에 하나의 파일씩 목록을 출력한다.

기타 다른 추가선택항목들에 대한 내용은 -help를 리용해서 알아볼수 있다. 또한 추가선택항목들은 그림 1-10과 같이 여러개를 동시에 사용할수 있으며 그 순서는 무관계하다.

그림 1-10. ls -al의 실행화면

그림 1-10에서 -a추가선택항목과 -l추가선택항목을 함께 주어 한 행에 하나씩 등록부의 모든 내용을 출력하도록 하였다.

pwd

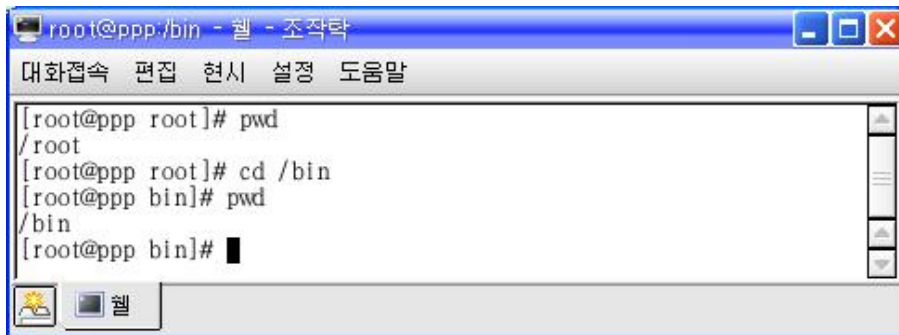



그림 1-11. pwd의 실행화면

현재 작업중인 등록부의 경로를 출력해준다.

pwd지령은 특별히 복잡한 등록부의 구조를 가진 작업공간에서 자기가 위치하고있는 등록부의 경로를 알아낼 때 유용하게 리용할수 있다.(그림 1-11)

cp

파일복사를 위한 지령이다. 기본문법은 다음과 같다.

$ cp 〔OPTION〕 원본파일 대상파일

그림 1-12와 같이 현재 등록부에 존재하는 bootlog.txt파일을 boot1.txt라는 이름의 파일로 복사해 보자.

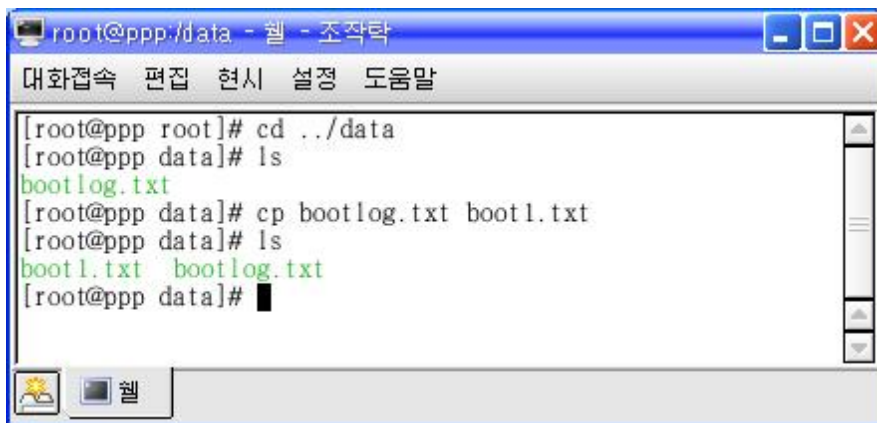



그림 1-12. cp의 실행화면

cp지령으로는 그림 1-13과 같이 현재 작업등록부로의 복사뿐만 아니라 다른 등록부로의 파일복사도 수행할수 있다.

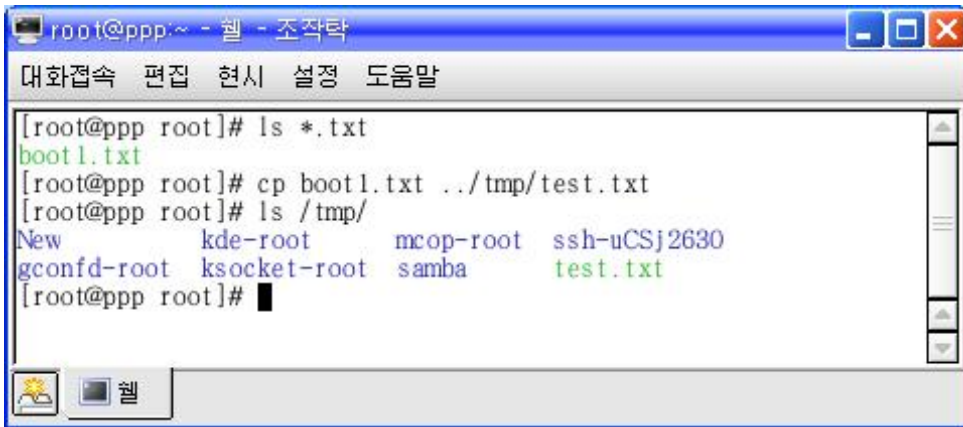



그림 1-13. 등록부로의 파일복사실행화면

그림 1-13과 같이 boot1.txt파일이 Temp/test.txt파일로 복사된 결과를 볼수 있다. 이 실례에서는 Temp등록부로 복사해야 할 상대경로를 사용하였다. 그러나 절대경로 역시 사용할수 있다. 다시 말하여 자기가 원하는 어떠한 등록부로도 파일복사를 수행할수 있다.

cp지령으로는 여러개의 파일복사도 지원한다. 그림 1-14에서는 boot1.txt, install.log파일을 Temp등록부로 복사하는것을 보여주고 있다.

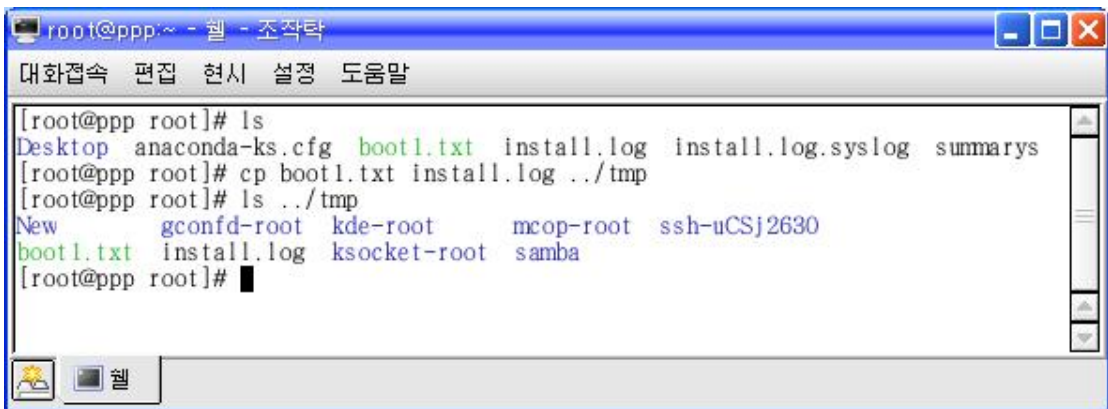



그림 1-14. 복수파일복사 실행화면

그림 1-15와 같이 《*》를 사용하여 여러개의 파일을 좀 더 쉽게 복사할수도 있다.

그림 1-15는 i로 시작하는 모든 파일을 Temp 등록부로 복사하는것을 보여주는 과정이다. 비슷한 문자렬을 가진 파일들을 복사할 경우 쓸모있게 사용할수 있다. cp지령은 표 1-6과 같은 여러가지 추가선택항목들도 지원한다.

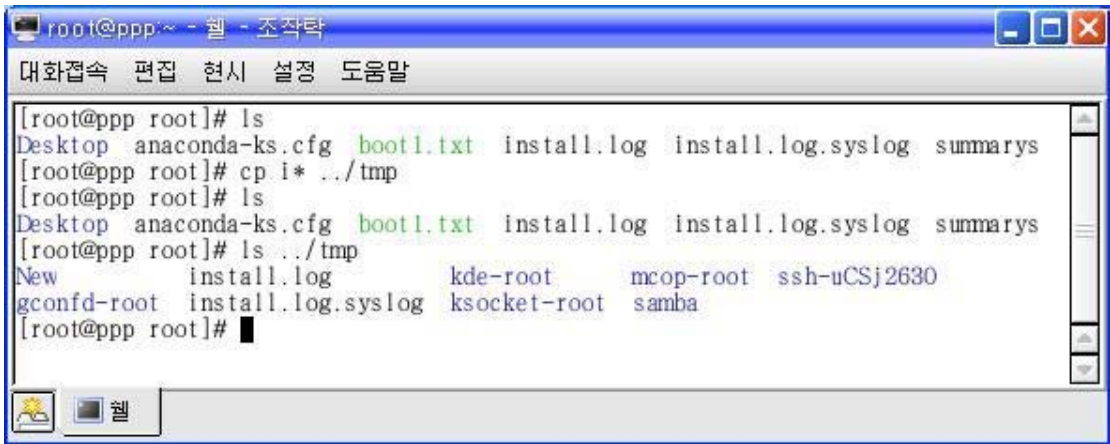



그림 1-15. *를 리용한 여러개의 파일복사

표 1-6. cp지령의 추가선택항목

추가선택항목	설 명
-a	-dpR추가선택항목과 동일한 추가선택항목
-b	파일을 삭제하기 전에 여벌복사를 진행한다.
-d	련결을 유지한다.
-f	이미 존재하는 대상파일을 질문없이 삭제한다.
-i	존재하는 대상파일의 덮어쓰기여부를 물어본다.
-l	파일을 복사하지 않고 련결을 한다.
-p	파일의 소유자, 그룹, 권한 등의 파일속성을 함께 복사한다.
-P	원본파일이 존재하는 등록부까지 복사한다.
-r	하위 등록부까지 복사한다.
-R	-r추가선택항목에서 련결기호(｜)나 /dev/zero와 같은 특수한 파일을 복사할 경우 쓰인다.
-s	파일을 복사하지 않고 기호련결을 생성한다.
-S	환경변수 SLMPLE_BACKUP_SUFFLX에 의해 설정된 여벌복사설정으로 여벌복사파일을 생성한다.
-u	원본파일이 복사될 파일보다 새것이거나 복사될 파일이 없을 때에만 복사한다.
-v	작업내용을 출력한다.
-V	일반적인 판본조종판(controler)을 겹쳐 쓴다.

mv

파일이나 등록부를 이동하거나 이름을 변경할 때 사용하는 지령이다. 문법은 다음과 같다.

$mv 〔OPTION〕 원본파일/등록부 대상파일/등록부

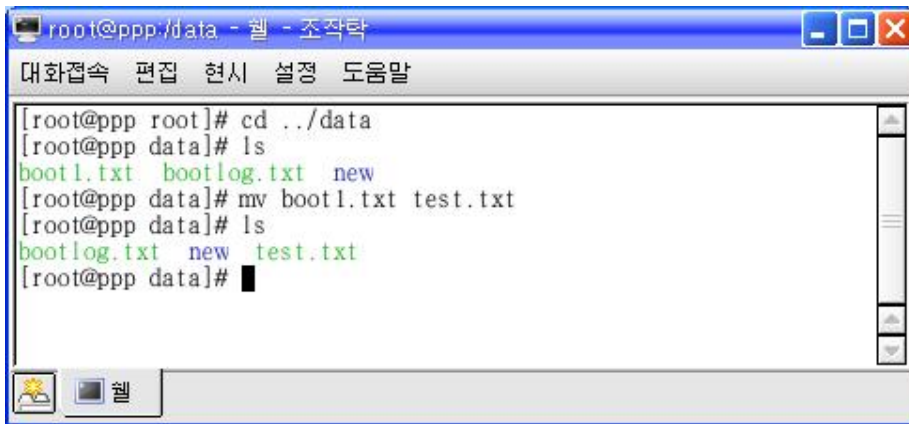



그림 1-16. mv를 리용한 파일이름의 바꾸기

우선 파일이나 등록부의 이름을 바꾸는 실행을 해보자. 그림 1-16은 boot1.txt를 test.txt로 바꾸는 과정을 보여준것이다.

mv는 파일이름뿐 아니라 등록부이름도 변경할수 있다. 그림 1-17과 같이 new라는 등록부를 directory라는 등록부로 바꾸어 보자.

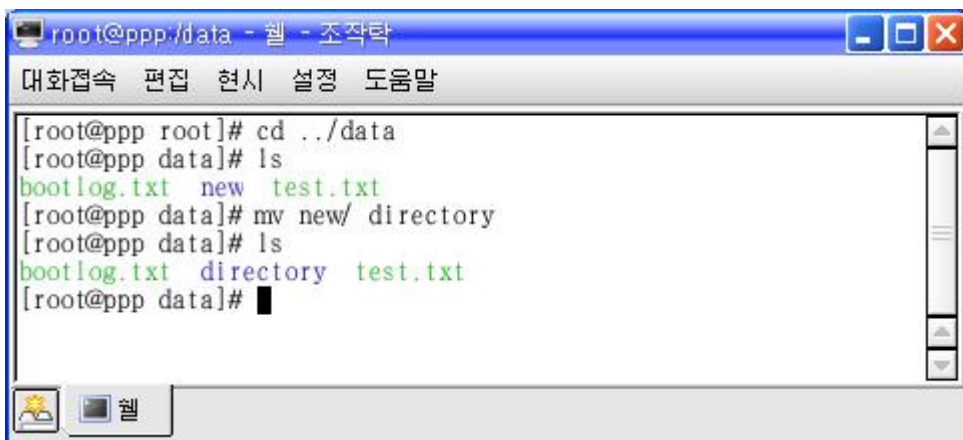



그림 1-17. 등록부의 이름바꾸기

mv는 다른 지령들과 달리 2가지 기능을 제공한다. 그 하나는 앞에서 설명한 파일 또는

등록부의 이름을 바꾸는것이고 또 다른 기능은 이제부터 설명할 파일과 등록부의 이동이다. 그림 1-18과 같이 test.txt파일을 directory등록부로 옮겨보자.

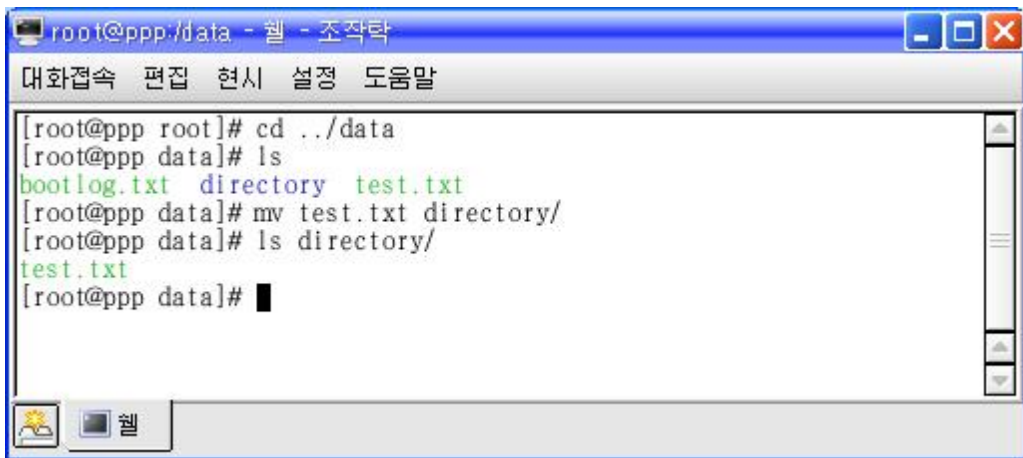



그림 1-18. mv를 리용한 파일의 이동

그림과 같이 파일이 옮겨질 등록부의 경로를 상대경로로 설정할수도 있고 절대경로로도 설정할수 있다.

mv는 또한 여러개의 파일을 동시에 이동시킬수 있다. 기본문법은 앞에서 본 cp지령과 같다.

표 1-7. mv지령에 추가할수 있는 추가선택항목

추가선택항목	설　명
-b	파일을 삭제하기 전에 여벌복사본을 생성한다.
-f	기존에 존재하는 대상파일/등록부를 질문없이 삭제한다.
-i	이동을 수행하기 전에 이동여부를 묻는다.
-S	일반적인 여벌복사머리부를 겹쳐 쓴다.
-u	오래된 파일과 새 파일만 옮긴다.
-v	이동작업을 출력한다.
-V	일반적인 판본조종기를 겹쳐 쓴다.

rm

파일을 삭제하는 지령이다. rm지령을 사용할 때 주의할 점은 Linux와 같은 Unix조작체계는 삭제된 파일의 복구를 지원하지 않으므로 신중하게 사용해야 한다. 기본문법은 아래와 같다.

```
$ rm [OPTION] FILE
```

실례로 그림 1-19와 같이 source.txt파일을 삭제해 보자.

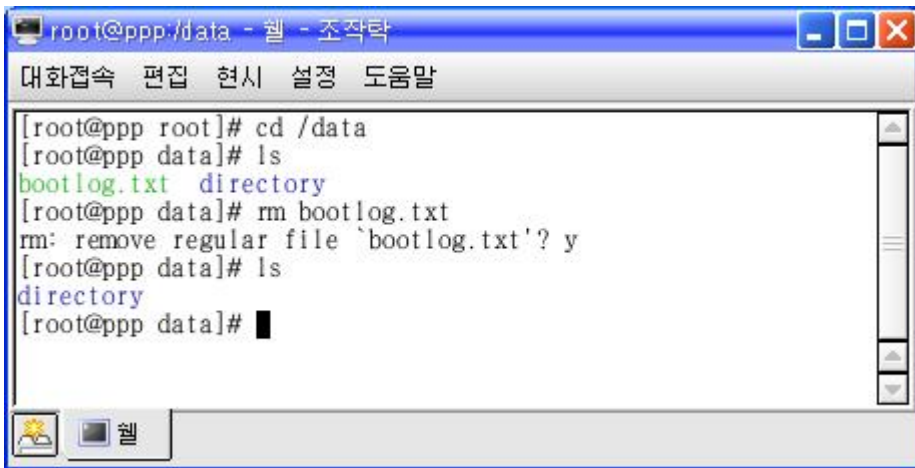



그림 1-19. rm지령을 리용한 파일의 삭제

rm지령을 실행하면 삭제한후 파일의 복구를 지원하지 않기 때문에 그림 1-19와 같이 다시 삭제여부를 묻는 통보문을 볼수 있다. rm지령은 《＊》를 사용한 여러개의 파일삭제도 지원한다.

그림 1-20과 같이 여러개의 파일을 삭제할 경우 삭제를 확인하는 물음에 일일이 대답하기가 시끄러우면 f추가선택항목을 주어 삭제에 대한 확인을 요구하지 않으면 된다. 그러나 다시 한번 강조하지만 Linux는 삭제한 파일에 대한 복구를 지원하지 않으므로 삭제에 신중하기 바란다.

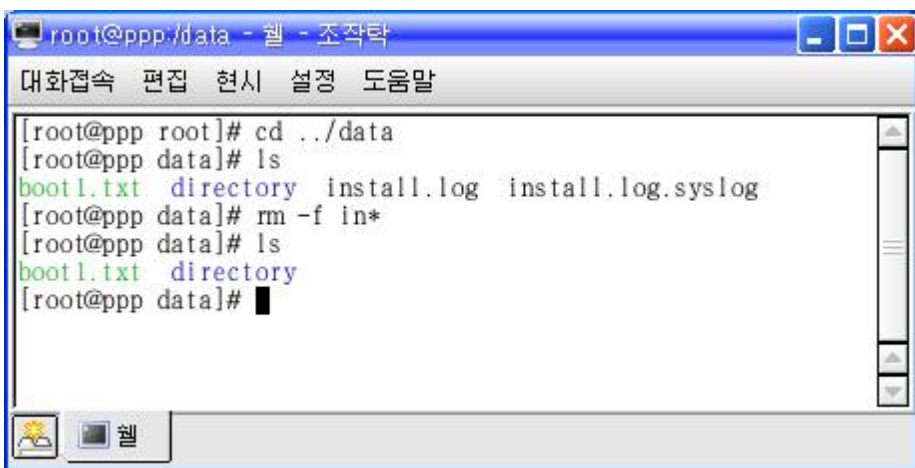



그림 1-20. 여러개의 파일지우기

기본적으로 rm지령은 파일삭제를 위한 지령이다. 하지만 r추가선택항목을 사용하면 등록부도 삭제할수 있다.(그림 1-21)

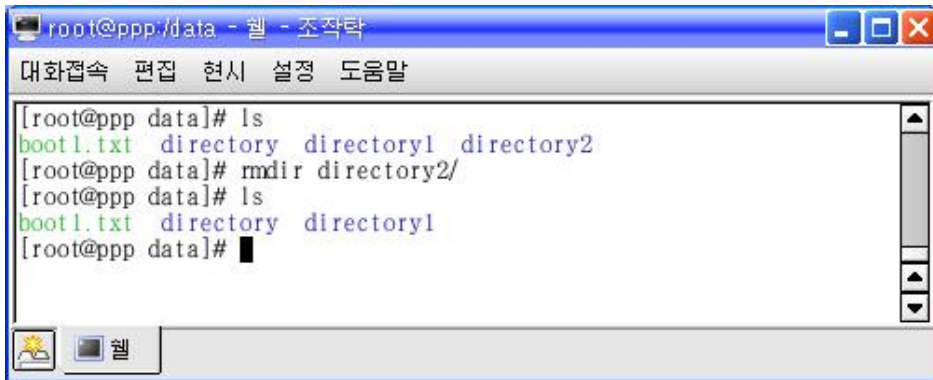



그림 1-21. 등록부의 삭제

그러나 여러가지 추가선택항목을 함께 사용하여 등록부를 삭제할 필요는 없다. 등록부의 삭제는 그림 1-21과 같이 등록부삭제지령 rmdir를 사용하면 된다.

표 1-8. rm지령에 추가할수 있는 추가선택항목

추가선택항목	설명
-d	등록부가 비여있지 않아도 등록부련결를 해체한다.
-f	삭제물음없이 파일을 삭제한다.
-i	삭제하기전에 삭제에 대한 확인물음을 한다.
-r	등록부의 내용을 재귀적으로 삭제한다.

mkdir

등록부를 생성하는 지령이다. 기본문법은 아래와 같다.

```
$mkdir 〔OPTION〕생성등록부이름
```

여기에 대한 설명은 그림 1-22와 같이 도움말을 통하여 알수 있다.

그러면 그림 1-23과 같이 DATA등록부상에 Directory등록부를 생성해보자.

등록부를 생성하면 그림과 같이 DATA등록부상에 Directory등록부가 생성된것을 볼수 있다. 다음으로 여러개의 등록부를 한번에 생성해보자.(그림 1-24)

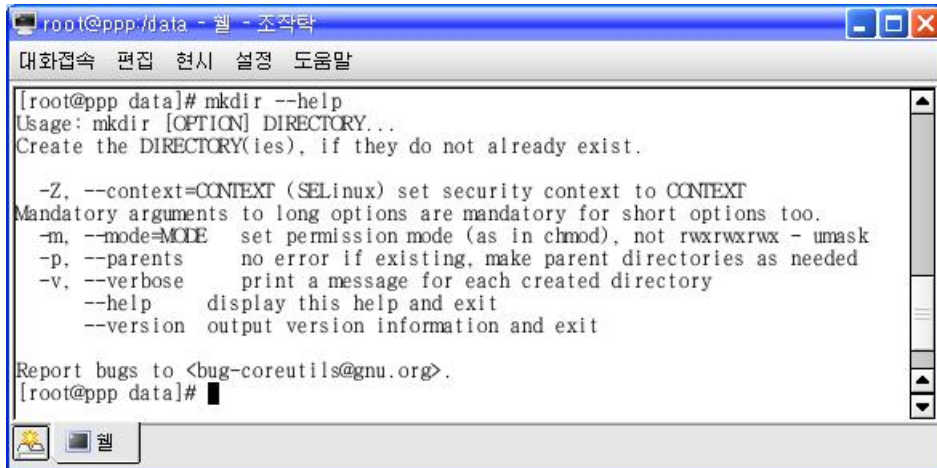



그림 1-22. 등록부 생성의 도움말

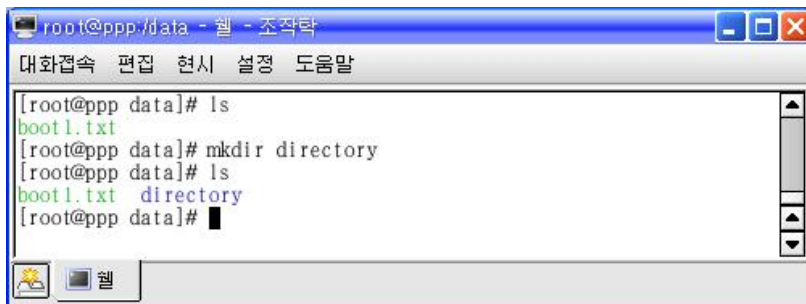



그림 1-23. 등록부의 생성

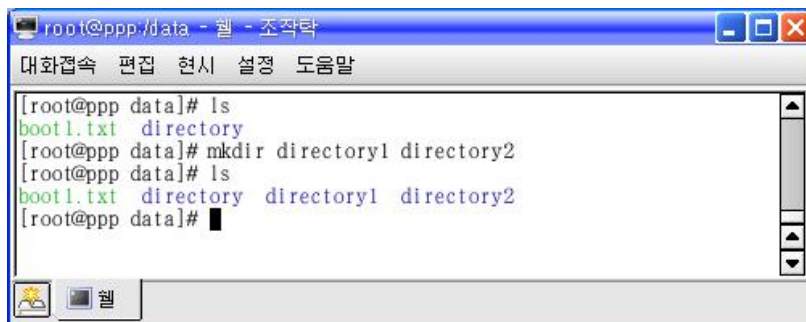



그림 1-24. 여러개의 등록부생성

mkdir지령으로는 등록부의 부분등록부도 한번에 생성할수 있다. 그림 1-25에 그에 대한 실례를 주었다.

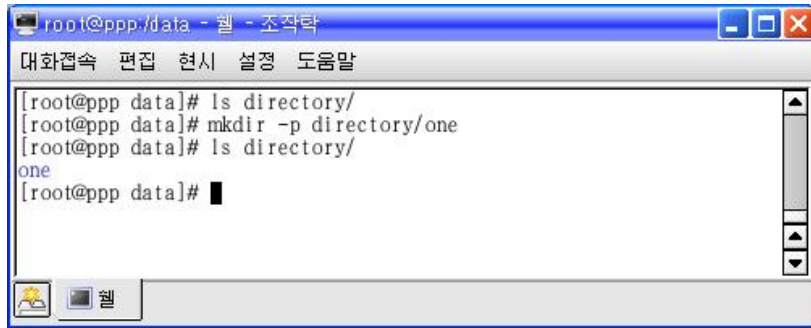



그림 1-25. 부분등록부의 생성

표 1-9. mkdir지령에 추가할수 있는 추가선택항목

추가선택항목	설 명
m	rwxrwxrwx- umask가 아닌 권한방식을 설정한다.
-p	필요한 경우 부분등록부도 생성한다.

man

지령이나 각종 자원들의 설명서를 출력하여준다. 기본문법은 아래와 같다.

```
$man〔OPTION〕 name
```

그러면 ls지령에 대한 설명서를 출력해보자.

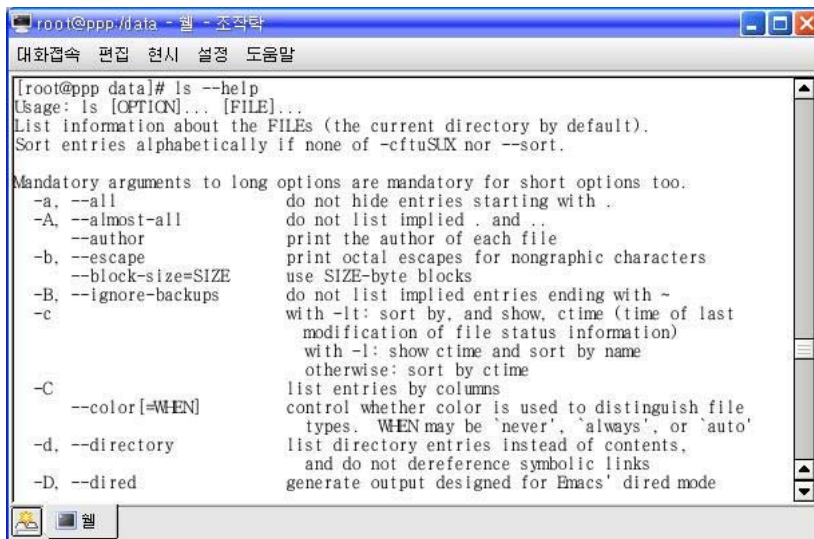



그림 1-26. 설명서의 출력

우와 같이 실행했다면 ls지령에 대한 설명서를 볼수 있을것이다.

설명서를 모두 출력할수 없을 경우에는 설명서를 나누어 출력하는데 space건을 사용하여 다음 페지로 이동할수 있다. 또한 설명서에서 빠져나오기 위해서는 ctrl+z건을 리용하면 된다.

표 1-10. man지령에 추가 할수 있는 추가선택항목

추가선택항목	설 명
-a	발견되는 모든 설명서페지를 출력한다.
-d	Debug정보를 출력한다.
-f	Whatis와 같다.
-h	도움말을 출력한다.
-k	apropos와 같다.
-t	groff사용하여 인쇄할수 있는 형식으로 설명서를 출력한다.

grep

주어진 실마리어를 포함하는 파일의 행을 출력한다. 기본문법은 아래와 같다.

```
$grep [OPTION] 실마리어 파일
```

그러면 nologin이라는 실마리어정보를 /etc/passwd파일에서 추출하는 과정을 실례를 들어 보기로 하자.(그림 1-27)

그림 1-27. grep의 실행결과

그림에서 볼수 있듯이 /etc/passwd파일에서 nologin을 포함한 모든 행을 출력한 결과를 볼수 있다.

표 1-11. grep지령에 추가할수 있는 추가선택항목

추가선택항목	설 명
-b	Byte묶음과 함께 출력한다.
-c	실마리어와 일치하는 문장의 행개수를 출력한다.
-C	실마리어와 일치하는 문장의 상하 두행을 포함하여 출력한다.
-h	파일이름은 출력하지 않는다.
-i	조사할 실마리어의 대소문자를 구분하지 않는다.
-l	실마리어와 일치하는 내용을 가진 파일이름만을 출력한다.
-n	실마리어와 일치하는 내용을 가지는 행번호를 출력한다.
-s	오유통보문만을 출력한다.
-v	실마리어와 일치하는 않는 행번호를 출력한다.
-V	판본과 함께 출력한다.
-w	한 단어로 일치해야 출력한다.
-x	실마리어와 전체 행이 일치해야 출력한다.

find

파일이나 등록부를 찾는 지령이다. 이 지령은 현재 등록부와 포함된 모든 부분등록부에서 파일이나 등록부를 찾는다. 기본문법은 다음과 같다.

그림 1-28. find지령의 실행결과

그림 1-28에서는 현재 등록부와 현재 등록부에 포함된 모든 부분등록부에서 test.txt 파일을 찾아 출력한 결과를 보여주었다. 최상위등록부인 《/》에서 찾기 시작하면 더 많은 파일들을 찾을수 있다.

echo

입력한 문자렬을 그대로 출력해주는 지령이다. 기본문법은 다음과 같다.

$echo〔OPTION〕 문자렬

그림 1-29와 같이 간단한 문자렬을 출력해보자.

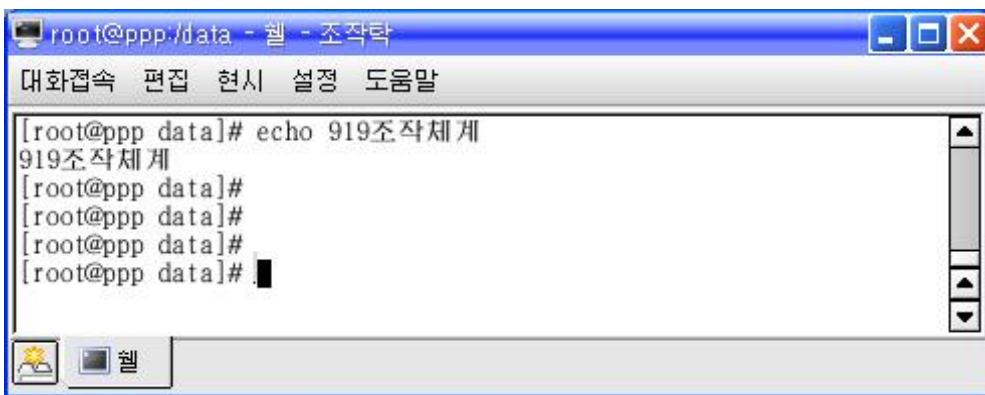



그림 1-29. echo지령의 실행실례

echo지령은 우의 그림과 같이 사용법이 간단하다. 하지만 일반적으로 echo지령은 여러 지령과 함께 조합해서 사용된다. 특히 쉘스크립트 등에서 유용하게 사용된다.

제3절. Linux조작체계의 등록부구조

Linux를 리용한 콤퓨터망의 사용에서 다음으로 중요한것은 봉사기를 잘 구축하는것이다. 관리하기 쉽고 운영이 안전하며 사용자들의 편리를 도모할수 있는 봉사기를 구축하는것은 결코 쉬운 일이 아니다.

여기에서도 Linux조작체계의 구조와 그의 동작원리를 잘 알고있는것이 무엇보다 중요하다. 자기가 운영하는 봉사기가 어떤 과정을 거쳐 어떤 파일들의 지원밑에 어떤 파일들을 만드는가에 대하여 알지 못한다면 유능한 봉사기관리자라고 말할수 없다는것은 누구나 쉽게 리해하리라고 본다.

Linux조작체계의 등록부구조 즉 파일체계의 구조는 Windows와는 다른 형태를 갖추고 있다.

Linux의 파일체계구조는 마치 나무가지와 같은 계층구조로 되여있다. 그래서 보통 Linux파일 체계구조를 나무(Tree)구조라고 한다. Linux의 파일체계구조는 기본적인 구조를 제외하고는 사용자의 설정에 따라 달라질수 있다. 하지만 대체로 Linux 파일체계의 표준안(FSSTND, Linux Filesystem standard)을 기반으로 하는것이 그중 좋다고 볼수있다. 이것은 Linux상에서 어떻게 파일체계를 구성할것인가에 대한 표준안을 제정하기 위해서 만들어진 문서이다. 이 표준안을 무조건 따라야 한다는 어떠한 규정은 없다. 다만 Linux파일체계표준안에 따라 파일체계를 구성하는 경우 파일들의 위치가 일관하게 유지되여 프로그람의 작성, 실행 및 체계관리도 쉽게 할수 있는 우점이 있다. 실지로 대부분의 Linux배포판들은 이 표준안을 지키고 있다.

Linux의 파일체계에서는 같은 목적의 파일들은 같은 장소에 일관하게 모아 관리한다. 례를 들면 지령어파일과 문서들이 제각기 독립된 장소에서 관리되고있다. 그렇다면 Linux파일체계가 어떻게 파일들을 관리하는가를 매개 등록부들을 보면서 자세히 보도록 하자.

일반적으로 뿌리(/)등록부라고 불리우는 등록부가 Linux조작체계에서 가장 최상위의 등록부이다. 체계관리자의 홈등록부인 /root등록부와는 다르다. 아래에 설명하게 되는 모든 등록부는 《/》등록부안에 위치하고있다.

/bin

Linux에서 가장 기본으로 되는 지령들이 모여있는 등록부이다. 해당 등록부안을 들여다보면 대부분의 파일들이 파일뒤에 '*'가 붙은 실행파일이라는것을 알수 있다.

특히 이곳에 위치하고있는 지령들은 체계의 기동에도 영향을 준다. 또한 기동후에 체계의 등록자리사용자들이 사용할수 있는 일반적인 지령들도 위치하고있다.

/etc

/etc등록부는 Linux조작체계에 관한 각종 환경설정파일들과 등록부들을 가지고있는 등록부이다. 이 등록부안에 있는 대부분의 파일들은 체계관리자에 의해서 관리되는 파일들이다.

웨브봉사기환경설정, 체계등록자리사용자정보, 통과암호관리, 체계의 파일체계관리파일, 여러가지 체계보안에 관련된 파일들, 체계초기화 설정파일, TCP/IP설정파일 등 체계전반을 포괄하는 거의 모든 환경설정파일들이 모두 이 등록부안에 보관되여있다.

- /etc/rc.d

체계의 기동과 체계실행준위변경시에 실행되는 스크립트들이 보관되여있는 등록부이다. Linux의 6가지 실행준위별로 매개의 해당 등록부가 있는것을 볼수 있을것이다.

- /etc/passwd

체계등록자리사용자들에 관한 각종 정보를 보관하고있는 사용자자료기지파일이다. 이 파일에는 해당 사용자의 ID는 물론 실제 이름, 암호화된 통과암호, 홈등록부, 사용자의 쉘정보 등 다양한 정보들이 보관되여있다.

- /etc/shadow

이 파일은 통과암호부분만을 따로 보관하는 파일이다. 이 파일의 통과암호는 암호화되여 Shadow통과암호 형태로 보관되여 있으며 체계관리자만이 접근할수 있기때문에 다른 사람들에게 공개될 우려가 적다.

- /etc/group

체계의 그룹에 대한 정보를 보관하고 있는 파일이다.

- /etc/inittab

init를 설정하는 파일이다.

- /etc/issue, /etc/issue.net

getty에 의해서 등록가입을 위한 지령재촉기호가 나타나기전에 출력되는 통보문을 설정하는 파일이다. Linux조작체계로 접속할 경우 가장 처음으로 볼수 있는 통보문이다. 보통 체계에 대한 설명과 각종 통보문를 전달하기 위해서 사용한다.

- /etc/motd

《Message Of The Day》의 략자로 체계에로의 접속에 성공할 경우 쉘이 동작하기전에 출력되는 통보문를 설정하는 파일이다.

- /etc/securetty

/etc/securetty파일에는 체계관리자가 체계에 등록가입할수 있는 안전한 사용자환경에 대한 정보가 보관되여 있다. 일반적으로 가상조종탁이 설정되여 있다. 이것은 콤퓨터망을 통해 체계로 침입해 체계관리자의 권한을 획득하는 비법사용자를 막기 위해서이다.

- /etc/shells

/etc/shells파일은 체계에서 안정적으로 사용할수 있는.(문맥검토) 쉘에 대한 정보를

보관하고있는 파일이다. 만약 chsh지령을 사용하여 사용중인 쉘을 바꾸려면 이 파일에 보관되여 있는 쉘중에 하나를 선택해야 한다. 또한 ftp데몬의 경우에는 사용자의 쉘을 검사하여 /etc/shells에 보관되여 있지 않은 쉘을 사용한다면 등록가입을 허용하지 않는다.

- /boot

Linux핵심부가 보관되여 있는 등록부로서 각종 Linux기동관련파일들이 보관되여 있는 등록부이다.

- /mnt

외부장치인 3.5″ 디스크구동기, CD-ROM구동기, Samba봉사기 등을 탑재하기 위해서 제공되는 등록부이다. 이 등록부는 림시로 사용되는 등록부이므로 프로그람들은 /mnt등록부에 어떠한 파일체계가 탑재되였는지 자동으로 인식하지는 않는다. 또한 /mnt는 보통 여러개의 하위등록부로 나누어 사용하며 평소에는 /mnt등록부의 매개 등록부들은 비여있는 상태이다.

- /usr

체계에 사용되는 각종 응용프로그람들이 설치되는 등록부이다. 거의 모든 응용프로그람과 그와 관련된 지령 및 서고들이 이 등록부에 위치하게 된다. 또한 X체계관련 파일들과 Linux핵심부원천파일, 각종 C언어관련 머리부파일 등도 이 등록부안에 보관되여 있다.

그러면 좀 더 자세히 /usr등록부안의 구조에 대해 보도록 하자.

- /usr/bin

Linux조작체계에서 사용되는 각종 프로그람들이 보관되여 있으며 /bin등록부에는 없는 다양한 실행파일들이 보관되여 있는 등록부이다.

- /usr/sbin

체계관리자를 위한 지령들을 보관하는 등록부이다. 보통 이 등록부에는 뿌리파일체계에 있을 필요가 없는 봉사기프로그람들이 보관된다.

- /usr/etc

/etc등록부에 각종 환경설정파일들이 있듯이 여기에도 여러가지 체계환경설정파일들이 보관되여 있다. 하지만 여기에 있는 파일들은 /etc등록부안의 파일들과는 달리 꼭 필요한 파일들은 아니다.

- /usr/include

C언어 관련 머리부파일들이 보관되여 있는 등록부이다.

- /usr/lib

각종 서고들이 보관되여 있는 등록부이다. 만약 자기가 직접 작성한 프로그람을 번역(compile)한다면 해당 프로그람은 /usr/lib등록부의 파일에 련결편집(link)된다. 또한 이 서고안에 실행코드가 필요하다면 /lib등록부를 참조한다.

- /usr/src

체계에서 사용되는 각종 프로그람들의 원천파일들이 보관되여 있는 등록부이다. /usr/src등록부에서 한번 살펴보아야 할 등록부는 /usr/src/linux등록부이다. 이 등록부는 Linux의 핵심부원천코드를 보관하고있는 등록부이다.

- /sur/X11R6

X-Window체계에 사용되는 모든 파일들이 이 등록부안에 보관된다. 이 등록부는 X-Window체계의 개발과 설치를 좀 더 쉽게 하기 위하여 전체 체계등록부구조에 통합되지 않고 독자적인 구조를 가진다.

- /usr/X386

/usr/X11R6등록부와 류사한 등록부로서 X11Release5를 위한 등록부이다.

- /usr/info

GNU Info문서들을 보관하고있는 등록부이다.

- /usr/doc

각종 문서들이 보관되여있는 등록부이다.

/lib

/lib등록부에는 체계에서 사용되는 각종 서고들이 보관된다. 특히 이 등록부에는 공동으로 사용되는 공동서고들이 보관된다.

/home

/home등록부에는 체계등록자리사용자들의 홈등록부와 ftp.www 와 같은 봉사등록부들이 보관된다. 이곳의 등록부와 파일들은 체계에서 사용되지 않는다.

/dev

/dev등록부에는 체계의 각종 장치들에 접근하기 위한 장치구동프로그람들이 보관된다. 등록부를 살펴보면 많은 파일들이 있는것을 볼수 있다. 하지만 이 등록부는 물리적인 용량은 가지지 않은 가상등록부이다. Linux조작체계는 Windows와는 달리 각종 장치들을 하나의 파일로 취급한다. 따라서 체계는 매개의 장치로부터의 정보를 /dev등록부에 있는 해당 장치파일로부터 가지고 온다.

- /dev/console

체계의 조종탁이다.

- /dev/hda

체계의 하드디스크이다. 여기서 /dev/had는 첫번째 하드디스크를 의미하는것이다. /dev등록부를 살펴보면 /dev/hda1, /dev/hda2 등의 파일들을 볼수 있을것이다. 이는 첫번째 하드디스크의 첫번째구획, 두번째구획 등을 의미한다. 만약 체계에 여러개의

하드디스크가 있다면 /dev/hdb,/dev/hdc 등의 파일들도 /dev/hda1, /dev/hda2 등과 같은 형식으로 보관되여 있을것이다.

- /dev/Ip

체계의 병렬포구장치들이다.

- /dev/null

/dev/null은 《검은 통》이라고도 불리우는 특별한 장치다. 이 장치에 자료 등을 보내면 모두 페기된다.

- /dev/pty

체계에로의 원격접속을 위한 《모조말단》이다. 만약 체계등록자리사용자들이 원격지에서 체계에로 telnet 등으로 접속을 시도한다면 이것들은 /dev/pty장치들을 사용하게 된다.

- /dev/sda

SCSI장치들이다. 만약 체계에 SCSI하드디스크를 장착했다면 체계는 /dev/hda 와 같은 파일들 대신 /dev/sda파일에서 정보를 얻어 장치하게 된다.

- /dev/ttyS, /dev/cuaS

/dev/ttyS는 직렬포구장치들이고 /dev/cuaS는 callout장치이다.

- /dev/tty

체계의 가상조종탁(Virtual Consoles)들이다. 이 가상조종탁의 기능은 하나의 화면에 여러개의 조종탁들을 만든다. 만약 사용자가 Linux체계를 기동시킨다면 Alt+F1, Alt+F2 등을 리용하여 Linux에서 제공하는 여러개의 가상조종탁(Virtual Consoles)을 직접 볼수 있을것이다.

/proc

/proc등록부에는 체계의 각종 프로쎄스, 프로그람정보 그리고 하드웨어적인 정보들이 보관된다. 이 등록부는 가상파일체계로서 /dev등록부와 마찬가지로 하드디스크상에 물리적인 용량은 가지지 않는다. 또한 이 등록부에 존재하는 파일들은 실제 하드디스크에 보관되지 않고 핵심부에 의해서 기억기에 보관된다. /proc등록부를 좀 더 자세하게 살펴보자.

- /proc/1

프로쎄스번호가 1인 프로쎄스에 대한 정보를 보관하는 등록부이다. 이는 곧 다른 프로쎄스번호를 가진 프로쎄스들도 사용자의 고유한 프로쎄스 번호의 등록부를 /proc등록부에 가진다는것을 의미한다.

- /proc/cpuinfo

프로쎄스의 정보를 보관하고있는 파일이다. CPU의 형, 모형, 제조회사, 각종 성능 등의 정보를 제공해준다.

- /proc/devices

현재 체계의 핵심부에 설정되여있는 장치들에 대한 정보를 보관하고있다. 이 파일로부터 체계의 모든 장치목록에 대한 정보를 얻을수 있다.

- /proc/dma

현재 체계에서 사용하고있는 DMA통로에 대한 정보를 보관하고 있는 파일이다.

- /proc/filesystem

체계에 설정되여 있는 파일체계에 대한 정보를 보관하고있는 파일이다.

- /proc/ioports

현재 사용중인 I/O포구에 대한 정보를 보관하고있는 파일이다.

- /proc/kcore

현재 체계에서 사용중인 기억기의 실제화상이다. 이 파일은 실제 기억기의 내용을 모두 가진것처럼 보인다. 그러나 이 파일은 프로그람이 필요로 하는 부분의 화상만을 필요한 때 만들어 제공한다.

- /proc/kmsg

핵심부에 의해서 출력되는 통보문들을 보관하고 있는 파일이다. 이것은 또한 syslog파일에도 보관된다.

- /proc/loadavg

현재 체계의 평균부하량(Load Average)에 대한 정보를 보관하고 있는 파일이다. 이 파일을 통해서 체계가 현재 수행해야 하는 일이 얼마나 많은가를 알려주는 3가지 지표에 대한 정보를 얻을수 있다.

- /proc/meminfo

현재 체계가 사용중인 기억기의 사용량을 보관하고 있는 파일이다. /proc/meminfo에서 실제 기억기는 물론 가상기억기에 대한 정보도 얻을수 있다.

- /proc/modules

현재 체계에서 사용중인 핵심부모듈에 대한 정보를 보관하고 있는 파일이다.

- /proc/net

체계의 콤퓨터망규약들에 대한 정보를 보관하고 있는 등록부이다. /proc/net정보를 통해서 현재 콤퓨터망규약들의 상태를 파악할수 있다.

- /proc/self

/proc/self등록부는 이 등록부를 사용하고있는 프로그람 자체의 프로쎄스등록부로 련결되여 있다. 그러므로 만약 서로 다른 2개의 프로쎄스가 /proc등록부를 보고있다면 두 프로쎄스는 서로 다른 련결을 보게 된다. 이를 통해서 프로그람들이 사용자의 프로쎄스 등록부를 쉽게 찾을수 있다.

- /proc/stat

체계의 현재 상태에 대한 다양한 정보를 보관하고 있는 파일이다.

- /proc/uptime

체계가 얼마나 오래 동작했는가에 대한 정보를 보관하고있는 파일이다.

- /proc/version

체계가 현재 사용중인 핵심부판본에 대한 정보를 보관하고있는 파일이다.

/var

/var등록부는 체계에서 사용되는 동적인 파일들이 보관되는 등록부이다. 각종 체계일지기록파일, 사용자 등록가입에 대한 보안기록 그리고 만약 체계에 우편봉사기를 운영한다면 사용자들에게 전송된 우편들을 림시로 보관한다.

/var등록부를 좀더 자세히 보도록 하자.

- /var/lib

체계가 동작하면서 계속 수정되고 변경되는 파일들을 위한 등록부이다.

- /var/local

/usr/local등록부에 설치된 프로그람들의 각종 자료들이 보관되는 등록부이다.

- /var/lock

잠금파일(Lock File)들이 보관되는 등록부이다. 일반적으로 많은 프로그람들이 특정한 장치나 파일들을 프로그람자신이 독점적으로 사용하려 할 때 /var/lock등록부에 잠금파일(Lock File)을 만들어 사용하게 된다. 때문에 다른 프로그람들은 장치나 파일을 사용하기전에 우선 /var/lock등록부의 내용을 조사하고 해당 장치나 파일들이 사용중인지 확인하게 된다.

- /var/log

프로그람들의 일지기록파일들이 보관되는 등록부이다. 이 등록부에서 중요한 파일은 wtmp와 message파일이다. 이것들은 각각 login과 syslog의 기록파일이다. wtmp는 체계의 모든 사용자등록가입과 가입탈퇴에 대한 정보를 보관하고 있는 파일이고 message는 핵심부와 체계의 모든 출력통보문을 보관하고있는 파일이다. /var/log안의 파일들은 체계의 사용량에 따라 그 크기가 무한대로 증가할수 있으므로 정기적으로 파일들을 삭제하는 등 등록부관리에 많은 주의를 돌려야 할 필요가 있다.

- /var/run

체계의 현재 정보들을 보관하고 있는 등록부이다. 간단하게 /var/run/atd.pid파일의 경우에는 현재 사용중인 atd데몬의 프로쎄스번호를 보관하고 있다.

- /var/spool

우편이나 새소식 등과 같은 체계상에서 대기상태에 있는 작업들을 위한 등록부이다. 개개의 대기작업들은 모두 /var/spool아래 고유의 등록부에 위치하게 된다. 례를 들어 체계의 등록자리사용자들의 우편은 /var/spool/mail에 보관되여 있다.

- /var/tmp

/tmp등록부에 보관된 림시파일들보다 오래 보관되여야 할 림시파일들이 보관되는 등록부이다.

/tmp

등록부의 이름에서도 알수 있듯이 림시파일들을 위한 등록부이다.

/ root

체계관리자의 홈등록부이다.

제2장. Apache봉사기

정보산업시대의 도래와 함께 사회생활의 여러 부문에 콤퓨터가 널리 리용되고 있지만 그중에서도 사람들이 가장 많이 리용하는것은 인터네트이다. 사람들은 인터네트를 리용하여 이전에는 상상도 할수 없었던 일들, 다시말하여 시간, 공간, 지리적위치 등의 개념들을 타파하고 전 세계를 자기의 활동무대로 삼게 되였다.

인터네트의 리용에서도 웨브는 가장 많이 리용되는 봉사로서 사람들속에서 널리 리용되고 있으며 오늘날에는 《인터네트=웨브》라고 이야기될 정도로 널리 리용되고있다.

이러한 웨브봉사를 가능하게 하여주는 봉사기들로는 여러가지가 있지만 그중에서도 Apache(아파치)봉사기는 80%이상의 사용률을 보여주는 거의나 독점적인 웨브봉사기이다. 이 장에서는 현재 세계적으로 널리 쓰이고있는 Apache봉사기의 설치와 사용방법을 실례를 통하여 취급하였다.

이 장의 간단한 구성체계는 아래와 같다.

- 웨브봉사기의 간단한 소개
- Apache봉사기설치와 등록부구조
- http.conf를 리용한 Apache봉사기의 환경설정
- 봉사기관리도구를 리용한 Apache봉사기의 구축과 관리

주의할점: 여기서 《Apache봉사기》라는 말은 Apache쏘프트웨어를 설치한 기대(콤퓨터)를 의미하지만 Apache쏘프트웨어를 념두에 두고 썼다.

제1절. 웨브봉사기의 간단한 소개

웨브(Web)봉사기에 대해 말할 때 사람들은 흔히 인터네트를 생각하게 된다.

인터네트라고 부르는 망(network)의 구성은 의뢰기와 여러가지 봉사들을 제공하는 봉사기들 [웨브, 전자우편(E-mail), FTP, IRC 등]의 련결로 볼수 있다. 이러한 봉사들중에서 웨브봉사는 가장 빠른 속도로 발전하고있으며 자기의 봉사령역에 다른 봉사기들의 기능을 포함하는 방향으로 나가고있다. 이로부터 웨브봉사기는 가장 중요한 봉사기로 인정되고있으며 널리 리용되고있다.

웨브봉사기에는 여러가지가 있다. 이중에서도 Apache봉사기(간단히 Apache)는 전세계적으로 놓고 볼 때 60%정도의 사용률을 보여주는 웨브봉사기시장에서 단연 첫째가는 등록자리를 차지하는 봉사기라고 말할수 있다.

Apache의 발전력사에 대해 간단히 본다면 이것은 1995년에 당시 가장 많이 사용되던 웨브봉사기인 NCSA를 기초로 하여 만들어졌다. 처음으로 개발되여 공개되였을 당시는 기능도 미미하였고 개선하여야 할 점들도 많았지만 계속되는 개발과정과 림시보수(Patch)과정을 거쳐 그 어디에도 뒤지지 않을 성능을 소유하게 되였으며 제일 큰 우점인 무료쏘프트웨어(free software)라는 점으로 하여 널리 사용되게 되였다. Apache라는 이름도 유래를 보면 NCSA웨브봉사기를 Patch했다고 해서 “A PATCH”의 조합으로 이름이 만들어졌다.

Apache의 기본적인 기능들을 보면 다음과 같다.

- 이미 널리 인정받은 보안성능과 안정성
- 원천의 공개
- UNIX계렬의 가동환경의 지원
- MS계렬(Windows 98/NT/2000/XP)의 가동환경지원
- 동적인 모듈기능의 지원
- 웨브봉사기의 세밀한 설정의 가능
- 가상콤퓨터의 지원
- HTTP1.1규약의 지원
- 간단한 인증으로부터 자료기지를 사용한 인증까지의 폭넓은 인증체계의 지원.

이러한 기본적인 기능들이외에도 제공되는 각종 모듈들을 사용하여 더욱더 강력한 기능을 지원하는 Apache웨브봉사기를 구축할수 있다.

 무료쏘프트웨어(Free Software)란 무엇인가?

주로 망우에서 자유롭게 배포되여 무료로 사용할수 있는 쏘프트웨어를 말한다. 무료쏘프트웨어는 무료이며 기능이 높고 단기능이지만 시장에서 판매되는 쏘프트웨어에서는 기대할수 없는 섬세한 요구에 대응한것도 많다.

류의해야 할 문제는 저작권을 포기하지 않기때문에 리용, 배포에 대하여서는 첨부되는 Read Me파일 등을 잘 읽고 저자의 의향을 따를 도덕적의무가 있다.

제2절. Apache봉사기의 설치와 등록부구조

Linux조작체계에서는 Apache봉사기의 설치가 아주 간단하며 쉽게 되여있다. 보통 Linux조작체계를 설치할 때 Apache봉사기의 설치를 선택하면 Apache봉사기가 설치된다. 그러나 필요에 따라 관리자가 직접 설치해야 하는 경우도 있다. 실례로 판본을 올려야 하는 경우라든가 관리도중 파일들에 이상이 생긴 경우 봉사기의 설치만 다시 하여야 한다.

앞에서도 설명한바와 같이 Linux조작체계에서 응용프로그람의 설치에는 두가지 방법이 있다. 하나는 원천패키지로부터의 설치이고 다른 하나는 RPM패키지로부터의 설치방법이다. 이 두가지방법은 Linux조작체계에서 거의나 공통적으로 적용되는 방법이다. 그러면 먼저 이 두가지 설치방법에 대하여 간단히 보도록 하자.

2.2.1. Apache봉사기의 설치

- 원천패키지로부터의 설치

먼저 Apache의 원천패키지압축파일을 구입하여 적당한 곳에 가져다 놓고 압축을 풀어야 한다. 압축을 푸는 과정은 아래와 같다.

```
[root@Linux temp]#tar xzvf
apache_1.3.23.tar.gz
```

압축을 푼 다음 아래와 같은 단계들을 거치면 Apache가 설치된다. 아래에서 - -prefix는 콤파일된 Apache를 설치할 등록부를 지정하는 선택항목이다. 이 항목을 지정하지 않으면 /usr/local/apache등록부에 설치가 된다.

```
[root@Linux apache_2.0.49]#./configure - -prefix=apache
[root@Linux apache_2.0.49]#make
[root@Linux apache_2.0.49]#make install
[root@Linux apache_2.0.49]#apache/bin/apachectl start
```

 봉사기와 의뢰기

간단히 말하여 봉사기(Server)란 봉사를 제공하는 콤퓨터를 말하며 의뢰기(Client)란 봉사기에서 제공하는 봉사를 받는 쪽을 말한다. 이전에는 장치개념에서 대형콤퓨터를 봉사기라고 이야기하기도 했지만 현재에는 이러한 개념이 별로 쓰이지 않고있다. 봉사기와 의뢰기는 상대적인 개념이다. 실례로 자기의 콤퓨터에 웨브봉사를 제공할수 있는 Apache가 설치되여있고 다른 사람의 콤퓨터에는 FTP봉사기가 설치되여있을 때 사용하는 용도에 따라 나의 콤퓨터는 Apache봉사기가 되기도 하며 FTP의뢰기로 될수도 있다. 여기서

configure는 설치할 프로그람을 콤파일하기전에 자기의 콤퓨터환경에 맞게 설정을 진행하는 지령이다. Make지령은 콤파일을 수행하도록 하는 지령이며 make install은 콤파일된 파일과 기타 여러가지 파일들을 지정된 등록부로 복사할 때 사용하는 지령이다.

여기까지가 Apache의 설치이다. 마지막에 준 지령은 Apache봉사기를 조종하는데 사용하는 지령으로서 아래의 표와 같은 선택항목을 리용하여 Apache를 조종할수 있다.

표 2-1. Apache봉사기 조종지령 선택항목

선택항목	동작과정
Start	Apache의 시작
Stop	Apache의 완료
Restart	Apache의 재시작
Fullstatus	Apache의 상태에 대한 모든 정보를 출력
Status	Apache의 상태정보를 출력
Graceful	Apache웨브봉사기의 재시작
Configtest	Apache웨브봉사기 설정파일의 문법검사

- RPM패키지로부터의 설치

RPM패키지를 적당한곳에 가져다 놓고 아래와 같이 RPM지령을 사용하여 설치하면 설치가 끝난다.

```
[root@Linux temp]#rpm    ivh apache_2.0.49-11.i386.rpm rpm.tar.gz
```

우에서 본것처럼 Apache봉사기의 설치는 매우 쉽다. 아래에서 언급되는 다른 기타 봉사기들의 설치도 거의나 마찬가지라고 볼수있다.

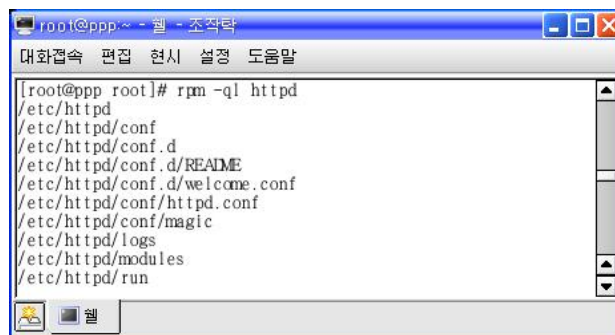



그림 2-1. Apache봉사기의 설치확인

Apache봉사기의 설치가 끝나면 그림 2-1과 같이 설치가 제대로 되였는가에 대하여 확인하여 본다. 이 과정에 설치의 정확성을 확인한다.

2.2.2. Apache의 등록부구조

우와 같은 방법들을 리용하여 Apache를 설치하였다면 관련파일들도 설치되였을것이고 그러면 Apache봉사기파일들이 어디에 설치되여있겠는가 하는 의문이 생긴다. 물론 어떤 파일들이 어떤 등록부에 들어가 있는가에 대해 왜 알아야 하는가에 대해서는 구태여 설명하지 않아도 리해하리라고 본다.

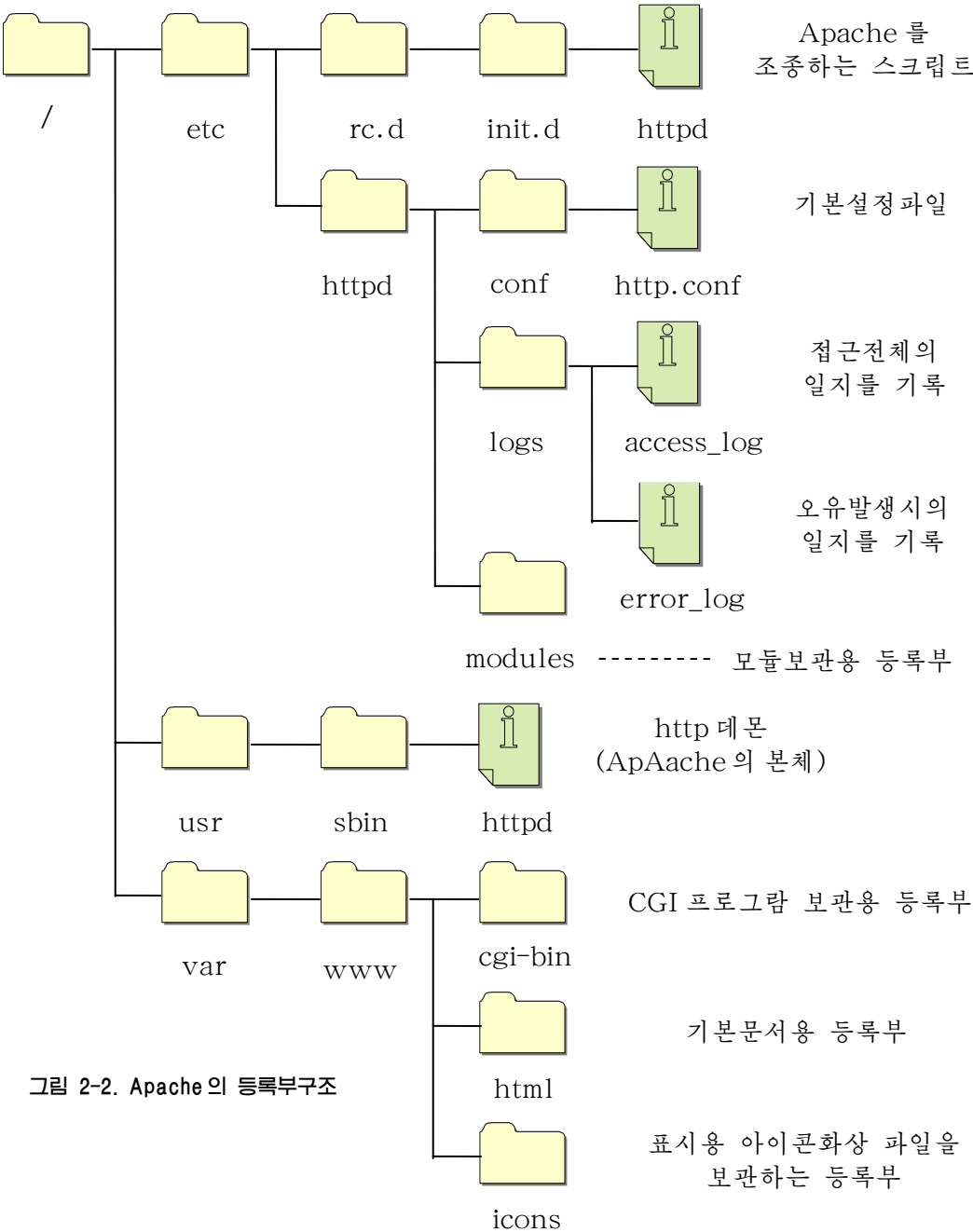



그림 2-2. Apache 의 등록부구조

그림 2-2는 Apache봉사기의 등록부구조이다. 매 등록부와 파일들의 사명과 역할에 대해서는 뒤에서 설명하므로 여기서 구체적인 설명은 략한다.

제3절. http.conf를 리용한 Apache봉사기의 환경설정

앞에서 우리는 Apache의 설치와 등록부구조에 대하여 보았다. 설치된 많은 등록부와 파일들 가운데서 가장 중요한 역할을 하는 파일은 </etc/httpd/conf/http.conf>라는 파일이다. 우에서도 간단히 언급한것처럼 이 파일은 Apache의 기본설정파일이다. Apache봉사기의 환경설정과 기타 많은 문제들을 이 파일에서 조종하게 되는것으로 하여 Apache에서 매우 중요한 파일로 되고있다. 실지 봉사기를 운영할 때에도 이 파일을 수정하거나 필요한 부분을 더 첨부해주는 방법으로 기본적인 봉사기의 동작환경을 조종하게 된다. 뒤에서 보게되는 응용프로그람을 리용한 봉사기의 시각적인 설치도 종당에는 이 파일에 기초하여 진행되게 되여있다.

이 파일은 크게 3가지부분 즉 대역환경(Global Environment)부분, 기본봉사기구성(Main server configuration)부분, 가상주콤퓨터(Virtual Hosts)부분으로 갈라서 볼수 있다.

참고로 이전 판본과의 차이를 말한다면 apache 1.3.4 이전 판본의 경우 3가지 파일(httpd.conf, srm.conf, access.conf)이 존재했지만 지금은 httpd.conf 파일로만 되여있는 상태이다.

그러면 매 부분의 구성과 그의 역할에 대하여 보도록 하자.

2.3.1. Global Environment부분

/etc/httpd/conf/http.conf파일을 편집기에서 열어보면 제일 먼저 있는 Global Environment부분을 볼수 있다. 여기에서는 많은 부분의 설정항목에 대하여 취급한다. 여기에서 나오는 설정항목들가운데서 중요한 부분에 대해서만 보도록 하자.

- ServerRoot부분

이 설정은 웨브봉사기의 설정파일, 오유파일, 일지기록(Log)파일이 보관되는 등록부를 지정하는 설정이다. 봉사기관리자가 보관하려고 하는 위치에 설정파일들을 보관하려고 한다면 이 부분을 수정하여야 한다.

```
# ServerRoot: 봉사기에서 사용되는 최상위등록부를 지정합니다.
# 그 밑에 봉사기구성파일, 오유 및 각종 기록 파일들이 놓입니다.
# 주의: 이것을 탑재된 nfs(혹은 다른 망) 상에 설정하려고 한다면 LockFile
  문서를 읽어보고 그것으로 인하여 생기는 오유들을 고려하여야 할것입니다.
# 등록부 경로를 밝힐 때 끝에 "/" 를 추가하지 말아야 합니다.
#ServerRoot "/etc/httpd"
```

우에서 실례로 보여준 부분이 표준적인 형식이다. 앞에서 설명한 Apache의 등록부구조와 련결하여 생각하여 보면 기본설정파일인 http.conf가 들어있는 conf등록부, 오유와 일지기록파일을 보관하는 logs등록부 등이 /etc/httpd등록부의 아래에 존재한다는것을 알수 있다.

- PidFile

```
# PidFile: 이것은 봉사기가 실행을 시작할 때 그의 프로쎄스 식별번호를 기록하는 파일을 지정하기 위한 지적자입니다.
#PidFile run/httpd.pid
```

PidFile부분에서 지정한 파일은 웨브봉사기가 실행될 때 사용자의 프로쎄스 식별번호를 기록하는 파일로서 일반사용자에게는 특별한 의미가 없다.

- ScoreBoardFile

```
# 이것이 지정되였다면 같은 이름을 가지는 scoreboard 파일이 없도록 하여야 합니다.
# scoreboard 파일은 국부디스크상에 저장되여야 합니다.
# ScoreBoardFile run/httpd.scoreboard
```

ScoreBoardFile부분에서 지정한 파일은 내부봉사기 프로쎄스의 정보를 보관하는 파일이다.

- Timeout

```
# Timeout: 요청을 받거나 보내기 전에 시간한계값을 초단위로 설정한것입니다.
#Timeout 300
```

Timeout는 의뢰기가 봉사기에 요청한 정보를 받을 때 소요되는 시간(받기와 보내기)을 설정하는 부분이다. 표준값은 우의 실례에서처럼 300으로 설정되여 있는데 초단위로 설정이 된다. 그러나 봉사기의 성능에 따라 필요한 시간만큼 정하여 주는것이 효률적이다.

- KeepAlive

KeepAlive는 지속성(persistent)접속을 사용할것인가 하지 않을것인가를 결정하는 지적자이다. 한번에 여러개의 접속을 통해 빠른 봉사를 제공할수 있게 해준다. 그러나 이것을 능동으로 설정하면 봉사기의 효률성이 떨어진다. 그 리유는 자식프로쎄스가 한가지 일만 처리하고 죽는것이 아니라 다음의 일도 처리함으로써 성능향상을 지향했기때문이다.

```
    #  KeepAlive:  이것은  의뢰기로부터의  지속적인  접속(persistent connection 이것은 접속에서 한개 이상의 요청이 있을 때를 의미한다)을 허용하거나 허용하지 않는것을 설정하기 위한 지적자입니다. 비능동으로 하기 위해서는  값을 "Off"로 설정합니다.
    # KeepAlive On
```

- MaxKeepAliveRequests

```
    # MaxKeepAliveRequests: 지속적인 접속을 허용하는 경우에  요청의 최대수를 지정하기 위한 지적자입니다. 무제한으로 하려면 0 으로 설정하면 됩니다. 실행능력을 최대한으로 하려면 이 값을 크게 하면 됩니다.
    # MaxKeepAliveRequests 100
```

MaxKeepAliveRequests지적자는 KeepAlive지적자를 on으로 설정한 경우 KeepAlive를 처리할수 있는 최대요청개수를 지정하는 지적자이다. 값을 0으로 하면 무한대가 되여 의뢰기가 접속을 끊을 때까지 계속 련결된 상태로 있다. 될수록 높은 성능을 기대한다면 이 값을 크게 설정하는것이 좋다.

- KeepAliveTimeout

```
  # KeepAliveTimeout: 같은 접속상에서 같은 의뢰기로부터 다음 요청이 올 때까지의 한계시간값을 초단위로 설정하는 지적자입니다.
  # KeepAliveTimeout 15
```

KeepAliveTimeout지적자는 KeepAlive지적자를 on으로 설정한 경우 같은 의뢰기에서의 같은 접속이 다음 요청을 할 때까지의 대기시간이다. 정해진 시간(단위는 초)안에 요청이 없으면 요청은 끊어지게 된다.

- MinSpareServer/MaxSpareServe**Error! Bookmark not defined.**

MinSpareServer, MaxSpareServer지적자를 설정하는 부분은 봉사기의 여유개수를 지정하는 부분이다. (여기에서 설정된 표준값은 대부분의 봉사기에 적합한 상태이다.)

Apache는 주기적으로 웨브봉사요청을 대기하고있는 웨브봉사프로쎄스의 개수를 확인하는데 설정값보다 작거나 크면 Apache는 새로운 프로쎄스를 생성하거나 제거한다.

즉 의뢰기의 요청이 많은 싸이트라면 적당히 이 값들을 증가 또는 감소해서 사용해야 한다. Apache는 정해진 최소값과 최대값을 통하여 의뢰기의 요청이 많은 경우든 적은 경우든 쉽게 프로쎄스를 늘이고 줄일수 있도록 한다. 다시말하여 봉사에 능동적으로 대처할수 있게 된다는것이다.

```
StartServers 8
MinSpareServers 5
MaxSpareServers 20
MaxClients 150
MaxRequestsPerChild 100
```

- StartServers

Apache봉사기의 초기 실행시에 여벌로 만들어지는 웨브봉사의 개수를 지정한다. 우의 실례에서 8이라고 설정하였으므로 모두 9개의 봉사기를 만들수 있다.

- MaxClients

MaxClients지적자는 동시에 웨브봉사기에 접속할수 있는 사용자수(의뢰기의 수)를 지정하는 지적자이다. 다시말하면 Linux조작체계에서 생성해낼 웨브봉사기프로세스의 최대값을 지정하는것이다. 이 값은 최대 512까지 리용가능하며 한명의 사용자당 3~4MB정도의 기억기가 필요하다고 추측하면 된다. 이 값을 늘이거나 줄일 경우 MinSpareServer, MaxSpareServer, StartServer의 값도 조절해주어야 한다. 다시말하여 최대값을 넘어설 때 의뢰기의 요청은 앞의 요청이 끝날 때까지 대기상태로 있어야 한다. 이 설정은 Apache봉사기가 많은 자원을 랑비하며 봉사기에 부하가 생기는것을 막기 위해 사용된다. (512이상의 값을 설정하려면 별도의 Pacth작업이 필요하다.)

- MaxRequestsChild

매개의 자식프로쎄스가 완료되기전까지 처리할수 있는 요청수를 설정하는 지적자이다. 대부분의 체계에서는 하나의 프로쎄스가 오래동안 실행되면 주기억기에 이상현상이 생기거나 또는 자원루출현상이 발생할수 있으므로 이것을 해결하기 위하여 웨브봉사기의 프로쎄스를 자동으로 완료하게 한다. 이 값을 설정하면 하나의 웨브봉사기프로쎄스는 자동완료전까지 지정한 값만큼의 요청을 처리하게 된다. 0으로 설정하면 자동완료는 진행되지 않는다.

- Listen

Listen지적자는 Apache를 특정한 IP주소 또는 포구와 결합시킬 경우 사용하는 지적자이다. 여기에는 다음 부분에서 설명하는 가상주콤퓨터도 사용할수 있다.

```
#Listen 12.34.56.78:80
Listen 80
```

- LoadModule

이 지적자는 Apache웨브봉사기를 설치할 때 동적공유객체(Dynamic Shared Object: DSO)기능이 지원되도록 콤파일했을 때에만 사용이 가능하게 된다. 기정값으로 설치하면 사용이 가능하다. DSO방식으로 만들어진 모듈의 기능을 사용하기 위해서는 먼저 이것을 사용하기전에 그 기능에 관련된 지적자를 LoadModule지적자로 선언해주어야 한다.

```
#LoadModule access_module modules/mod_access.so
 LoadModule auth_module modules/mod_auth.so
 LoadModule auth_anon_module modules/mod_auth_anon.so
 LoadModule auth_dbm_module modules/mod_auth_dbm.so
...
```

주의할 점은 이전 판본(1.3.20이전)에서는 이 설정이 AddModule과 짝을 이루어야 사용되였으므로 LoadModule지적자의 설정이 수정되면 AddModule지적자의 설정도 수정해야 한다는것이다. 따라서 자기가 사용하는 판본에 대하여 잘 알아보고 필요한 추가와 삭제를 진행하여야 한다.

지정하는 방식은 앞의 실례와 같이 《LoadModule 모듈이름, 모듈위치》의 순서로 지정해야 한다. 정적으로 콤파일된 모듈들은 여기에서 지정해주지 않아도 된다. 정적으로 콤파일된 모듈들은 httpd -l지령을 리용하여 확인할수 있다.

참고로 이야기 한다면 이 설정은 놓이는 순서가 매우 중요하게 작용한다는것이다. 따라서 사용자모듈을 새로 추가할 때에는 반드시 가장 마지막부분에 추가하여야 한다.

- ExtendedStatus

```
#ExtendedStatus On
```

이 지적자는 Apache에서 server-status처리자가 호출되였을 때 Apache의 자세한(전반적인) 봉사기상태정보를 만들것인가 아니면 기본적인 상태정보만 만들겠는가를 설정하는 지적자이다. 표준값은 off이다.

2.3.2. 기본봉사기구성(Main server configuration)부분

이 부분은 가상주콤퓨터가 아닌 기본봉사기(main server)의 설정들을 관리하는 부분이다. 그러나 이 부분에서 설정한 값들은 가상주콤퓨터에 영향을 준다. 그러므로 Apache봉사기의 설정과 관리에서 매우 중요한 부분이라고 말할수 있다.

여기에서 서술된 지적자들은 아래에서 취급하는 가상주콤퓨터부분에서 사용될수 있으며 이 경우 가상주콤퓨터는 여기에서 정의한 값이 아니라 자체로 정의한 값을 사용한다. 여기에 대해서는 다음 부분에서 자세히 보도록 하자.

이 부분에서는 다음과 같은 지적자들에 주의를 돌려야 한다.

- User/Group

웨브봉사기가 실행되는 권한을 지정하는 지적자이다. 일반적으로 nobody권한으로 웨브봉사기를 실행한다. nobody로 지정하면 웨브봉사기를 통해 실행되는 모든 CGI나

프로그람들은 nobody권한으로 실행되게 된다. 만약 nobody로 설정되여진 상태에서 보안을 철저히 하려면 telnet으로부터의 접근을 막아야 한다.

일부 핵심부들은 그룹의 값이 60000을 초과할 때 setgid나 semctl을 거절한다. 이러한 체계들에서는 Group #-1을 사용하지 말아야 합니다.

```
#User nobody
Group nobody
```

- ServerAdmin

```
#ServerAdmin root@localhost
```

ServerAdmin은 웨브봉사기의 관리자전자우편주소를 설정하는 지적자이다.

간단히 말하여 봉사에서 문제가 생겼을 경우 전자우편을 보낼 주소를 적어주는곳이라고 리해하면 된다

- ServerName

```
#ServerName ppp.sec.edu:80
```

ServerName은 봉사기가 자기 자체를 확인하기 위하여 사용하는 이름이나 포구를 설정하기 위한 지적자이다. 여기에는 사용자의 등록된 DNS를 적어주면 된다.

이것은 대체로 자동적으로 결정되지만 기동시에 문제가 발생하지 않도록 명백히 설정해주는것이 좋다.

주콤퓨터에 대해서 이것이 정확한 DNS이름으로 설정되지 않았다면 봉사기측에서 진행하는 방향바꾸기(redirections)가 진행되지 않는다.

만일 DNS이름이 등록되지 않았다면 봉사기의 IP주소를 입력하여도 된다. 이때 그 주소로는 어떤 방법으로든지 접근할수 있어야 하며 정확한 방법상에서 redirection이 이루어져야 한다.

- Document Root

```
DocumentRoot "/var/www/html"
```

Document Root는 웨브봉사기에 봉사요청이 들어오면 제공할 HTML파일들을 보관해두는 등록부를 지정해주는 지적자이다. 다시말하여 웨브봉사기의 root등록부가 된다.

실례를 들어 설명한다면 웨브열람기가 http://servername/index.html파일을 요청하면 봉사기는 이 Document Root지적자에서 지정한 등록부에 보관된 index.html을 보내주게 된다.

- <Directory 경로>, </Directory>

<Directory 경로>는 경로에서 지정한 등록부의 권한에 대한 설정을 할 때 사용하는 지적자이다. 이 지적자는 HTML표쪽(tag)문처럼 <Directory 경로>로 시작하여 </Directory>로 마감을 해주어야 한다. 여기서 설정한 값들이 영향을 받는 범위는 <경로>에서 지정한 등록부의 안에 있는 등록부까지이다.

```
<Directory "/var/vhost/html">
        Options Indexes Includes FollowSymLinks
        AllowOverride None
        Allow from from all
        Order allow,deny
</Directory>
```

여기서 사용한 지적자들에 대하여 보기로 하자.

Option지적자는 《Option 추가선택항목이름 추가선택항목이름 . . .》의 구조로 되여있다.

많이 쓰이는 추가선택항목의 종류와 그의 역할은 다음과 같다.

- None - 아무런 추가선택항목도 적용하지 않는다.
- All - MultiViews를 제외한 모든 값을 적용한다.
- Indexes - 등록부를 보여주는 추가선택항목으로서 해당 등록부안에 index.html파일이 없을 경우 등록부와 파일들을 목록화하여 보여준다.
- Includes - SSI(Server Side Includes)의 실행을 가능하게 한다.
- ExecCGI - CGI응용프로그람을 실행할수 있도록 등록부를 설정하는것이다.

AllowOverride지적자는 .htaccess파일에 놓일수 있는 지적자들을 위한것이다. 이 지적자의 뒤에 놓이는 추가선택항목이름이 None일 경우 .htaccess내용이 access.conf의 추가선택항목에 의해 효력이 없어지게 된다.(해당 등록부안에 .htaccess파일이 존재할 경우)

즉 access.conf설정의 어느 부분을 무시하겠는가의 여부를 설정한다.

Order지적자는 Directory지적자를 통하여 설정하는 경로에 접근하는 접근권한을 설정할 때 사용하는데 이때 먼저 적용받으려고 하는 규칙에 대하여 지정한다.

Allow지적자는 Directory지적자를 통해 설정하는 경로에 접근하는 접근권한을 설정할 때 접근을 허용할 의뢰기의 정보에 대하여 지정할 때 사용한다. 주콤퓨터나 령역 또는 전체의 허용 등이 가능하다.

Deny지적자는 Directory지적자를 통해 설정하는 경로에 접근하는 접근권한을 설정할 때 접근을 거부할 의뢰기의 정보에 대하여 지정할 때 사용한다. 주콤퓨터나 령역 또는 전체의 허용 등이 가능하다.

- Ifmodule

Ifmodule지적자는 지정된 모듈이 사용가능할 때 이 지적자로 둘러쌓인곳에 설정해놓은 추가선택항목들을 적용하기 위하여 사용한다. 때문에 Apache웨브봉사기의 실행시에

지정된 모듈이 발견되지 않으면 Ifmodule지적자로 둘러싸인곳에 설정된 추가선택항목들도 적용이 되지 않는다.

```
<IfModule prefork.c>
    StartServers 8
    MinSpareServers 5
    MaxSpareServers 20
    MaxRequestsPerChild 100
</IfModule>
```

- DirectoryIndex

```
DirectoryIndex  index.php index.html index.htm index.shtml
```

이 지적자는 DirectoryIndex로 사용할 파일을 지정할 때 사용하는 지적자이다. 다시말하면 어떤 등록부에 접근했을 때 가장 먼저 읽어야할 파일을 지정하는것이다. 보통 여기에는 index.php, index.html, index.htm, index.shtml 등을 지정하며 여러개인 경우에는 우의 실례와 같이 공백으로 구분한다.

- AccessFileName

```
#AccessFileName .htaccess
#<Files ~ "^\.ht">
    Order allow, deny
    Deny from all
</Files>
```

매 등록부에 대하여 접근조종정보내용을 담고있는 파일의 이름을 지정하는 지적자이다. 우의 실례에서 지정한 파일과 함께 《.ht》로 시작되는 파일들은 접근권한설정에 관계되는 파일들이다. 그러므로 이 파일들은 보안에 직접적인 영향을 주는 파일들이므로 웨브에서의 접근이 불가능하여야 한다. 때문에 우의 두번째와 같이 설정을 하면 지정된 파일들은 웨브봉사기를 통해 접근할수 없게 된다.

여기서 Files지적자는 우에서 본 Directory지적자와 같은 역할을 수행한다.

- DefaultType

```
#DefaultType text/plain
```

이 지적자는 봉사기에서 사용되는 문서중 MIME형으로 알수 없는 문서에 대해 적용할 기본 MIME 형을 지정할 때 설정한다.

표준값은 text/plain이며 이 값은 주로 봉사에 문서파일이 많을 경우에 사용한다. 만일 2진파일이나 실행파일이 많을 경우에는 이 추가선택항목을 application/octet-stream으로 써넣으면 된다. 이것은 웨브열람기가 실행파일 등을 본문문서로 착각하고 열람기에 표시하지 않기 위해서이다.

- TypesConfig

```
#TypesConfig "/etc/mime.types"
```

이 지적자는 MIME형의 정보가 기록된 파일의 위치를 지정할 때 사용하는 지적자이다.

- HostnameLookups

```
#HostNameLookups Off
```

이 지적자는 Apache가 의뢰기의 요청을 받아들이면서 IP를 확인하고 확인한 주소를 DNS봉사기에 다시 의뢰하여 방문자의 주소를 기록할것인가 안할것인가를 지정하는 지적자이다. 요청을 진행하면 DNS봉사기에 한번 더 의뢰하기때문에 상당한 시간이 걸리며 봉사기에 부하가 걸리게 된다. 때문에 필요한 경우가 아니면 표준값인 off를 사용하는것이 좋다.

- ErrorLog

```
ErrorLog "/var/log/httpd/error_log"
```

웨브봉사기의 동작과정이나 웨브봉사중에 발생한 오유통보문을 보관할 파일(일지기록파일)을 지정하는 지적자이다.

<VirtualHost>지적자부분에서 ErrorLog설정을 하지 않았다면 여기서의 설정을 따른다. 일지기록파일의 경우에 시간이 흐르면 계속 용량이 증가하기때문에 수시로 검사하여 보고 용량을 줄여야 한다.

- LogLevel

```
#LogLevel warn
```

Log정보를 기록할 때 기록의 수준을 결정하는 지적자이다. debug, info, notice, warn, error, crit, alert, emerg 등의 값을 지정할수 있다.

- LogFormat

이 지적자는 CustomLog지적자에서 사용할 몇가지 형식에 대한 별명(Aliaas)을 정의한다. 여기에서 사용하는 추가선택항목들의 의미는 다음과 같다.

%h - 주콤퓨터명
%l - 원격일지기록 이름
%u - 사용자 인증에 사용된 사용자이름
%t - 시간
%r - 요청한 내용의 첫번째 줄
%s - 봉사기의 상태
%b - 전송량(머리부포함)
%{머리부} - 요구된 머리부의 내용
%U - 요구한 URL

우의 의미는 common, referer라는 별명으로 매개의 log형식을 정의한 실례이다.

```
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
…
```

- CostomLog

```
# CustomLog 파일위치 형식
CustomLog logs/access_log combined
```

CustomLog는 사용자정의 일지기록형식으로 일지기록파일을 기록할 때 사용하는 지적자이다.

<VirtualHost>지적자부분에 이 설정부분이 없을 경우 여기서 설정한 위치에 모두 보관된다. 가상주콤퓨터별로 따로 일지접근기록파일을 만들고 싶으면 이 설정을 매개의 <VirtualHost>지적자부분안에서 해주면 된다.

- ServerSignature

```
# Set to one of:  On | Off | EMail
ServerSignature on
```

이 지적자는 봉사기가 생성하는 오유페지나 FTP등록부목록, mod_status를 사용한 출력페지에 봉사기의 판본과 주콤퓨터의 이름행을 추가할 때 사용하는 지적자이다. 만일 이 값을 Email로 설정하면 ServerAdmin지적자에서 지정한 전자우편주소를 표시하게 된다.

- Alias

```
# Alias 가상등록부 “실제경로”
Alias /icons/ "/var/www/icons/"
```

Alias지적자는 웨브봉사기의 문서뿌리등록부가 아닌 다른 경로의 등록부를 마치

웨브봉사기의 문서등록부처럼 사용하려고 할 때 지정하는 지적자이다. 지금까지의 설명에 의하면 Apache봉사기에서는 여러가지 리유로 하여 웨브봉사기의 root등록부나 그 안의 등록부가 아니면 접근이 불가능하게 되여있다. 그러나 필요에 따라 어떤 봉사에서는 접근을 허용하여야 하는 등록부가 있게 된다. 이럴 때에는 이 지적자를 리용하여 사용이 가능하게 하도록 하여야 한다.

우의 실례대로 한다면 웨브봉사기의 "/var/www/icons/" 등록부를 /icon등록부로 알고 봉사를 받도록 하게 될수 있다.

참고로 알아둘것은 이 지적자를 제대로 사용하자면 mod_alias.so모듈을 선언해주어야 한다는것이다.

- ScriptAlias

```
# ScriptAlias 가상등록부 "실제경로"
ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"
```

ScriptAlias지적자도 Alias지적자와 마찬가지로 가상등록부를 만들 때 사용되는 지적자로서 Alias와 다른점이 있다면 일반문서파일이 아닌 CGI와 같은 실행가능한 스크립트파일이 보관되여있는 가상등록부를 설정할 때 사용된다는 점이다. 즉 이 지적자로 설정된곳에 있는 파일에 대해서는 Apache봉사기가 실행을 시도하게 된다.

가상주콤퓨터에서 이 지적자를 사용하려면 가상주콤퓨터별로 다음과 같은 설정을 해주어야 한다.

```
<Directory "/var/www/cgi-bin">
        Options ExecCGI
        AllowOverride None
        Allow from from all
        Order allow,deny
</Directory>
```

- AddLanguase

```
#AddLanguage da .dk
AddLanguage nl .nl
…
```

이 지적자는 봉사기에서 사용되는 문서의 언어를 표시한다. 다시말하여 열람기가 리해할수 있는 언어를 나타낸다.

한가지 알아둘 점은 뽀르뚜갈어는 Perl언어와 구별하기 위하여 pl.po로 표시한다는것이다.

- LanguagePriority

이 지적자는 문서에서 문장을 읽어들이는 동안 여러가지 언어가 존재할 경우 언어의 우선권을 설정해주는 지적자이다.

설정은 배렬된 순차로 우선권을 가진다.

```
#LanguagePriority en da nl et fr de el it ja kr no pl pt pt-br ltz ca es sv tw
```

- AddHandler

```
#AddHandler cgi-script .cgi
```

이 지적자를 사용하면 특정한 파일확장자와 취급기(Handler)를 련결하거나 특정한 파일의 형식에 특정한 동작을 련결할수 있다.

우에서 제시한 실례의 뜻은 .cgi로 끝나는 모든 파일을 ScriptAlias와는 상관이 없이 CGI로 인식하여 실행하라는 의미이다.

아래에서 대표적인 실례와 그의 동작을 보자.

HTML 파일 사용

AddType text/html .shtml

AddHandler server-parsed .shtml

Apatch의 send-asis HTTP 파일 기능 사용

AddHandler send-as-is .asis

봉사기처리 imap 파일을 사용

AddHandler imap-file .map

Type map을 사용

AddHandler type-map .var

- AddType

```
#AddType application/x-tar .tgz
```

이 지적자는 mime.types에 지정되여 있지 않는 형식을 추가할 때 사용하는데 지정한 형식에 해당한 취급자가 필요할 때에는 우의 AddHandle과 같이 사용한다.

- ProxyServer

이 부분은 대리봉사기의 사용을 설정하는 부분이다. 대리봉사기모듈을 포함했을 경우 #를 제거하면 대리봉사기의 사용이 가능하게 된다.

```
#<Proxy *>
    Order deny,allow
    Deny from all
    Allow from .your-domain.com
#</Proxy>
```

- ProxyVia

```
#Set to one of: Off | On | Full | Block
#ProxyVia On
```

이 지적자는 HTTP/1.1 <Via:>머리부를 처리하겠는가 안하겠는가를 설정하는 부분이다.

추가선택항목에는 Off | On | Full | Block이 있는데 Full은 봉사기의 판본정보를 포함하며 Block는 나가는 모든 자료에서 <Via:>를 제거한다.

- Cache

```
#CacheRoot "/etc/httpd/proxy"
#CacheSize 5
…
```

캐쉬기능을 활성화하기 위해서는 이 지적자들의 #를 없애면 된다.

2.3.3. 가상주콤퓨터(Virtual Hosts)부분

《하나의 체계에서 두개이상의 웨브싸이트를 봉사할수는 없을가?》

봉사기를 어느정도 다루어본 사람들이라면 어느정도 이런 생각을 가져본 사람들이 있을것이다. 이 질문에 대한 대답을 이제 설명하려는 가상주콤퓨터기능에서 찾아보기로 한다.

실제로 하나의 독립적인 체계에서 하나의 웨브봉사기만을 운영한다면 그것은 엄청난 장치구입비와 운영비부담을 전제로 하게 된다. 모든 공장, 기업소, 기관, 학교들이 자기의 웨브싸이트를 가지려고 하는 경우에도 이것은 천문학적인 수자에 달하게 된다. 거기에 중요한 기관들인 경우 여러개의 싸이트를 가져야 할 때가 있는데 이것도 역시 난문제로 제기되는것이다.

이러한 문제를 해결하기 위하여 가상주콤퓨터기능을 리용한다.

가상주콤퓨터란 물리적으로 하나인 독립적인 체계에 매개가 서로 다른 령역(domain)이름을 가진 여러개의 웨브싸이트를 운영하는 방법을 말한다.

가상주콤퓨터의 방식은 크게 두가지로 볼수 있는데 하나는 IP주소를 리용한 가상주콤퓨터이고 다른 하나는 Name주소를 리용한 가상주콤퓨터방식이다.

이 두방식의 차이는 거의 없다고 볼수 있다. 때문에 여기서는 IP주소를 리용한 가상주콤퓨터에 리용되는 지적자에 대하여 설명한다.

먼저 망기판에 여러개의 IP주소를 부여해준다.

실례로 eth0:0, eth0:1을 사용하여 여기에 IP를 부여한다.

여기서는 기본 IP를 192.168.8.54로 하고 가상주콤퓨터의 IP를 192.168.8.154로 설정하였다.

- VirtualHost

```
<VirtualHost 192.168.8.154>
ServerAdmin pak@ppp.sec
ServerName ppp.sec
DocumentRoot /var/vhost/html
DirectoryIndex index.php index.html index.htm index.shtml
</VirtualHost>
```

이 지적자는 가상주콤퓨터에서 사용할 전반적인 설정을 해주는 부분이다. IP와 NAME방식의 차이점은 이 지적자의 뒤에 IP를 적어넣는가 NAME을 적어넣는가가 다를뿐이다. 안에서 사용된 지적자들은 우에서 이미 설명하였으므로 설명을 생략한다. 이 뒤에 가상주콤퓨터에서 사용할 여러가지 지적자들을 적어주면 가상주콤퓨터의 설정은 끝나는것이다. 아래에 실례를 주었다.

```
<Directory "/">
        Options FollowSymLinks
        AllowOverride None
</Directory>

<Directory "/var/vhost/html">
        Options Indexes Includes FollowSymLinks
        AllowOverride None
        Allow from from all
        Order allow,deny
</Directory>
…
```

외에도 많은 지적자들이 있지만 지금까지 설명한 지적자들이 봉사기의 설정에서 제일

중요한 부분들이라고 볼수 있다. 설명한 부분들에 대한 설정이 끝나면 Apache봉사기의 설정이 끝났다고 볼수 있다. 봉사기의 설정이 끝났다면 한번 확인해볼 필요가 있을것이다. 확인은 매우 간단하다. 설정파일을 정확히 수정하였는가를 다시 한번 확인하여 보고 conf파일을 기억시키고 Apache데몬을 재시동한다. 다음 망으로 련결된 콤퓨터에서 mozilla라든가 Internet Explorer와 같은 열람기를 기동시키고 해당한 IP와 NAME을 주어 아래의 그림들과 같이 응답이 오면 Apache의 설정이 제대로 된것이다.

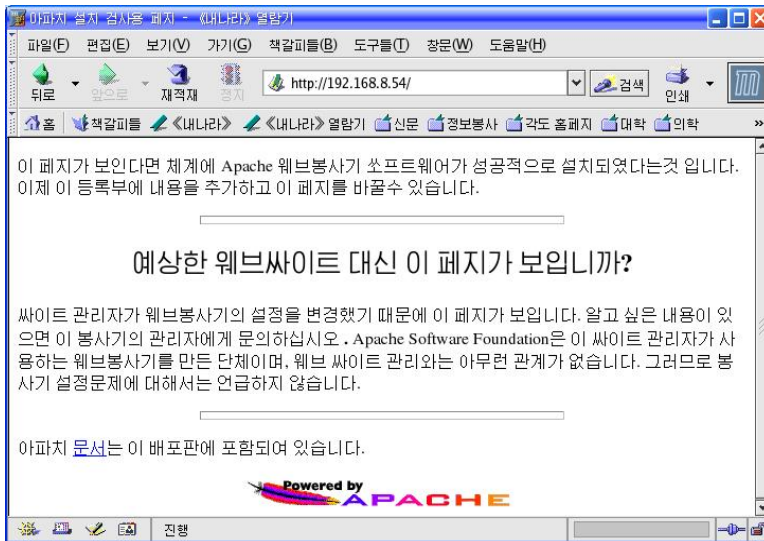



그림 2-3. Apache봉사기의 기동

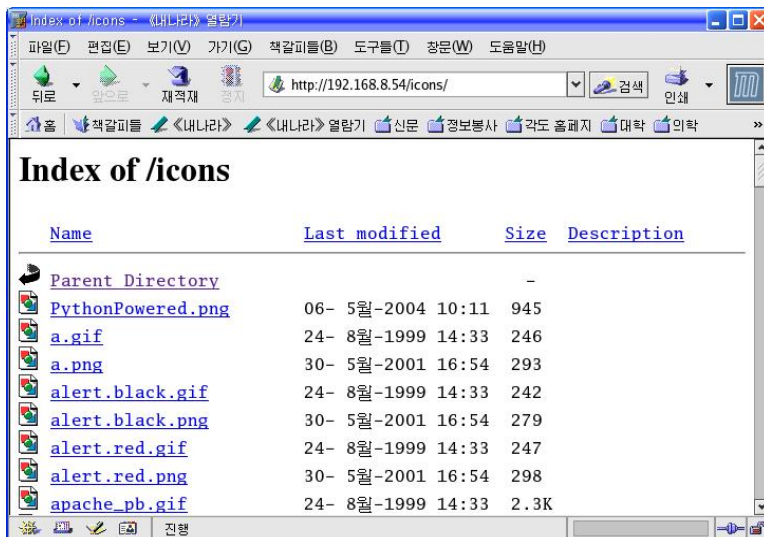



그림 2-4. Alias지적자를 리용한 가상등록부의 설정

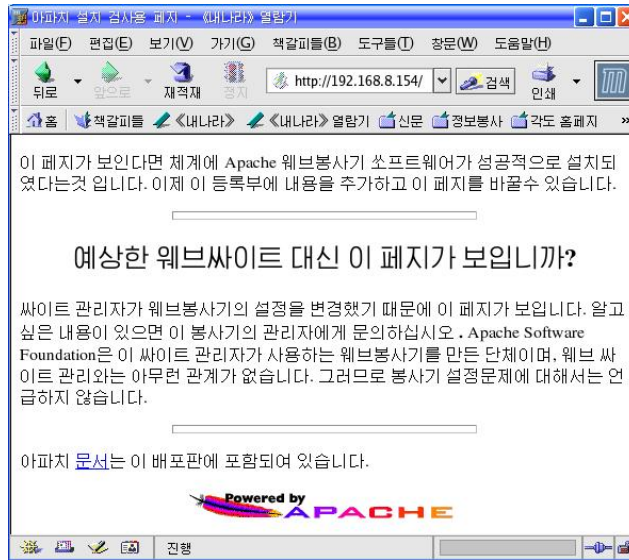



그림 2-5. 가상주콤퓨터를 리용한 192.168.8.154봉사기의 실행

 IP주소란?

TCP/IP규약으로 통신을 진행할 때 송신자와 송신선을 구별하기 위하여 할당하여 주는 주소를 말한다.

일반적으로 8bit마다 점으로 구분된 4개의 10진수로 표시된다. 실례로 192.168.8.14로 표시되는 주소를 말한다.

전체 길이는 32bit이며 종류는 기본적인 클라스 A,B,C,D 그리고 E의 5가지가 있다.

Windows조작체계에서는 Internet Explorer에서 확인하여 보면 된다.

Windows조작체계에서도 우와 같은 결과가 나왔다면 Apache봉사기의 설정이 제대로 되였으며 봉사기가 제대로 동작한다는것을 말한다.

제4절. 봉사기관리도구를 리용한 Apache봉사기의 구축과 관리

2.4.1. 봉사기관리도구의 소개

여러가지 봉사기들의 설치와 구축, 관리의 기본과정은 지금까지 설명한 Apache봉사기의 설치과정과 거의나 류사하다고 볼수있다. 그런데 이전의 봉사기의 구축은 앞에서 설명한것처럼 구축파일들을 직접 조작하고 수정해야 하므로 봉사기의 구축을 처음해보는 초학자들에게는 상당히 어려운 문제로 제기되였다.

물론 구축파일들을 직접 수정하는것이 자기의 특성에 맞는 봉사기를 구축하고 관리하는데서 제일 좋은 방법인것만은 사실이다. 그러나 이 방법은 숙련과 구축과정에 대한 깊은 지식을 요구한다. 이러한 문제점들로 하여 봉사기의 구축과 운영은 몇몇 한정된 사람들 일명 봉사기관리자라고 하는 사람들에게만 국한된 문제로 인식되여왔다.

그러나 최근 인터네트가 폭발적인 인기를 얻으며 사회생활의 여러부문에서 널리 리용되고 있는 현실은 누구나 콤퓨터망을 구축하고 활용할것을 요구하고 있다. 이러한 요구를 만족시켜주기 위하여 봉사기의 구축과 관리를 보다 쉽게 할수 있도록 하는 여러가지 도구들이 나오고 있다. 이 절에서는 우리 식 조작체계에서 지원하여 주는 봉사기관리도구에 대해서 취급한다.

봉사기관리도구의 실행은 조작체계의 대면부에서 《시작》단추를 누르고 거기로부터 《응용프로그람-인터네트-봉사기관리도구》의 순서로 봉사기관리도구를 찾아서 실행시키면 된다.

그림 2-6과 2-7은 봉사기관리도구의 대면부이다.

그림 2-6. 봉사기관리도구(환경설정편)

그림 2-7. 봉사기관리도구(봉사기설정편)

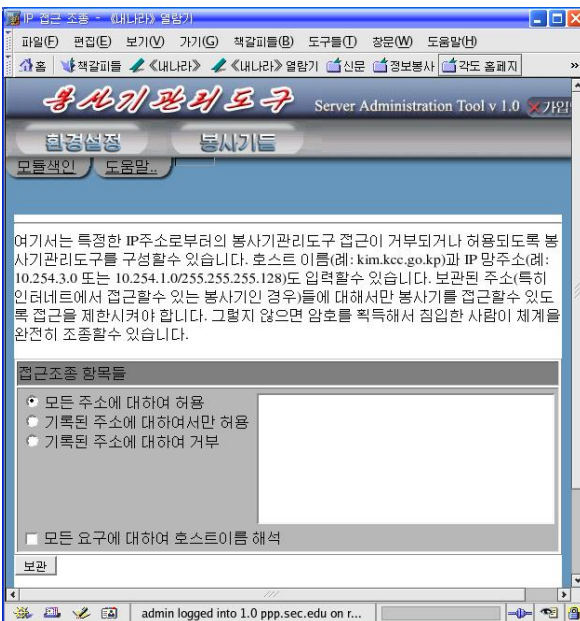



그림 2-8. IP접근조종항목의 대면부

그림들로부터 알수 있는바와 같이 봉사기관리도구를 리용하면 봉사기의 구축과 관리 그리고 환경설정을 쉽게 진행할수 있다는것을 알수 있다.

봉사기들의 구축과 관리에 대해서는 아래로 내려가면서 매 봉사기마다 설명을 진행하므로 여기서는 환경설정부분에 대해서만 간단히 보기로 하자.

환경설정부분은 그림에서도 알수 있는것처럼 IP접근조종, 포구 및 주소, 사건기록, 언어, 인증, SAT사용자관리라는 6개의 항목들로 갈라져있다.

이 6개의 항목들로 봉사기의 환경설정을 진행한다.

매 항목에 대해서 간단히 보도록 하자.

그림 2-8은 IP접근조종항목의 대면부이다.

그림에서도 보고 알수 있듯이 이 항목에서는 봉사기가 접근을 허용하거나 거부하여야 하는 주소에 대해 설정하는 항목이다.

이 설정항목은 봉사기의 관리에서 매우 중요한 항목인데 그것은 여기에서 설정해준 원칙에 따라서 의뢰기들의 접근을 허용하거나 거부하기 때문이다. 그러므로 IP접근조종항목의 설정을 잘 하는것은 봉사기의 보안에서 기초로 된다.

그림 2-9는 사건기록의 대면부이다.

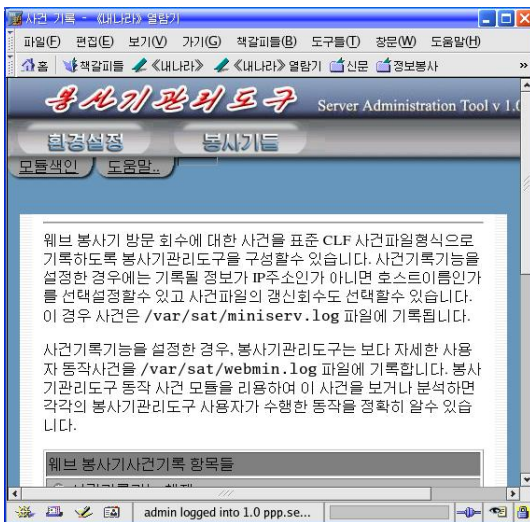



그림 2-9. 사건기록항목의 대면부

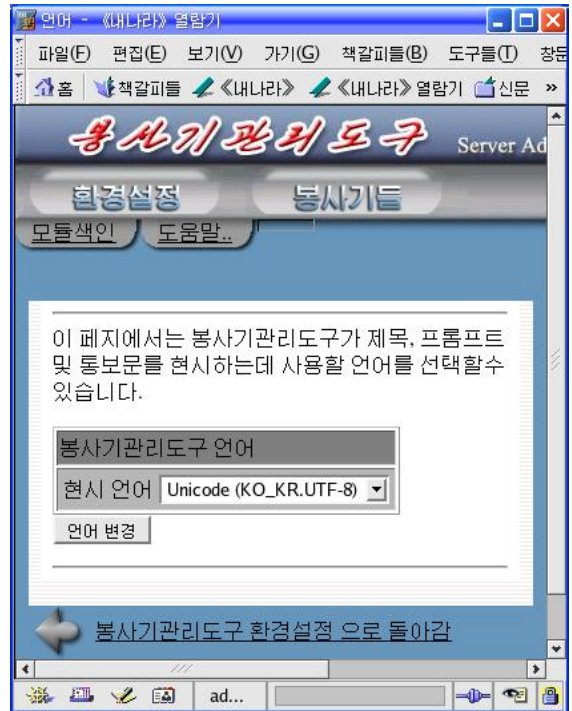



그림 2-10. 언어설정항목의 대면부

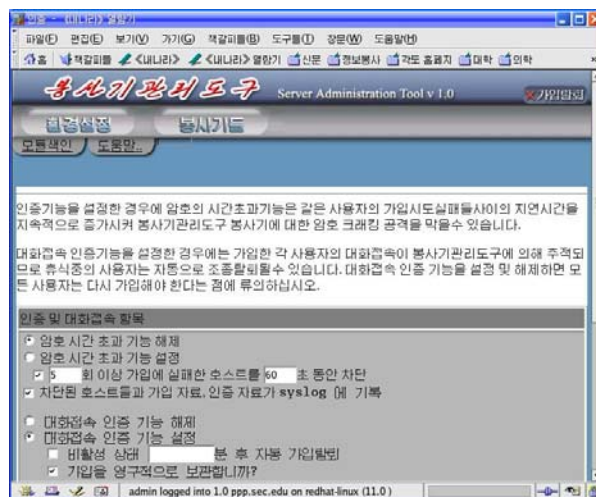



그림 2-11. 인증항목의 대면부

사건기록은 앞에서도 설명한 일지기록파일(log파일)을 관리하는 항목이라고 말할수 있다. 일지기록파일의 관리는 매우 중요한 의의를 가지는데 그것은 이 파일을 리용하여 해당한 의뢰기접근의 경로와 동작에 대한 자세한 정보를 얻을수 있기때문이다. 일지기록파일의 분석을 정확히 하여야 해당 봉사기에 열려져 있는 《비법적인 구멍》들과 그를 리용한 침입흔적을 분석하여 봉사기를 안전하게 관리할수 있기때문이다. 또한 봉사기의 오유정보에 대한 분석을 정확히 진행할수 있다.

언어항목은 그림 2-10에서 알수 있는것처럼 제목이나 통보문현시에서 사용할 언어를 설정하는 항목이다. 간단히 말하면 언어설정항목이라고 할수 있다.

그림 2-11은 인증항목의 대면부이다. 인증항목을 리용하면 봉사기에 대한 접근시 인증기능의 설정과 통과암호시간초과기능과 같은 기능들을 리용하여 봉사기에 대한 해커공격과 불법침입을 막기 위한 설정을 효과적으로 할수 있다.

이상과 같은 항목들을 리용하여 봉사기의 설정과 관리를 효과적으로 진행할수 있다.

2.4.2. 관리도구를 리용한 Apache봉사기의 관리

우에서도 간단히 언급한것처럼 관리도구를 리용하여 봉사기들을 관리할수 있다.

그러면 관리도구를 리용한 Apache봉사기의 설정과 관리에 대하여 보도록 하자.

그림 2-7에 나오는 봉사기항목에서 《Apache봉사기》항목을 찰칵하면 아래의 그림과 같은 창이 나온다.

여기에는 앞에서 설명한 Apache봉사기관리를 위한 필요한 항목들이 아이콘으로 나타나있다. 필요한 항목들에 대해 간단히 보도록 하자.

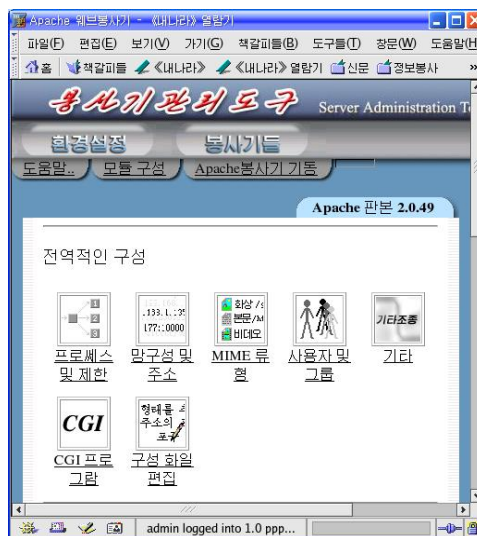



그림 2-12. Apache봉사기관리대면부

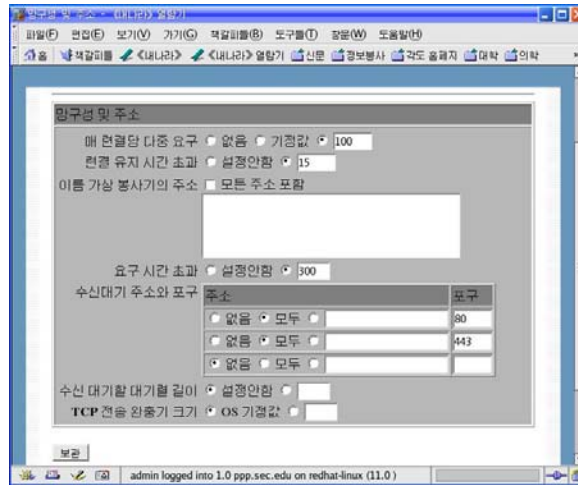



그림 2-13. 망구성 및 주소를 관리하는 대면부

그림 2-14. 프로쎄스 및 제한을 관리하는 대면부

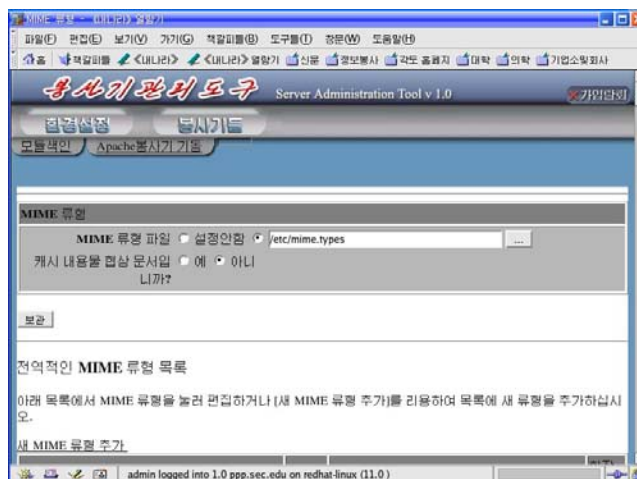



그림 2-15. MIME류형설정항목의 대면부

그림 2-16. 사용자 및 그룹 설정항목 대면부

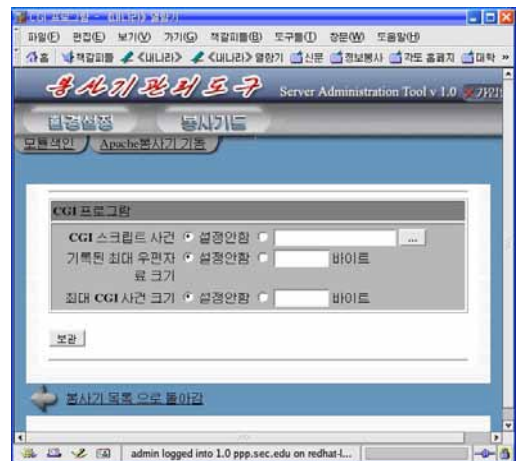



그림 2-17. CGI프로그람 설정항목 대면부

그림 2-18. 기타 항목의 설정대면부

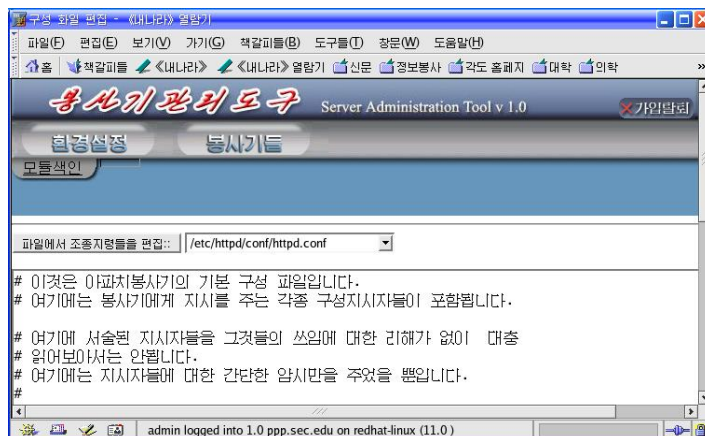



그림 2-19. httpd.conf파일편집대면부

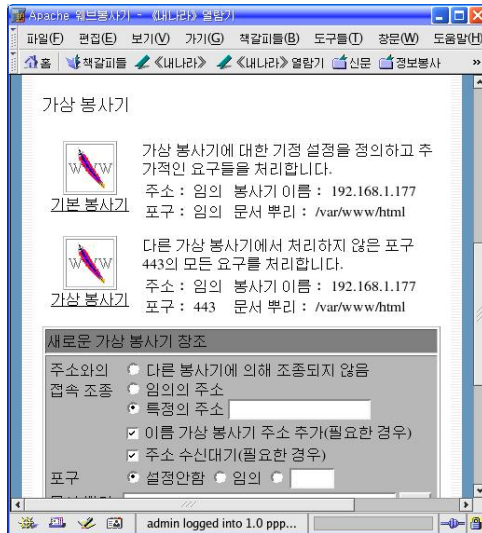



그림 2-20. 가상봉사기의 설정대면부

우의 그림들을 보면 앞에서 설명한 httpd.conf파일에서 설명한 부분들이라는것을 알수 있다.

간단히 말하여 봉사기관리도구도 종당에는 변화된 대면부의 자료를 가지고 httpd.conf파일을 변화시켜 봉사기에 대한 설정과 관리를 진행한다. 단지 이러한 도구를 리용하는것은 봉사기의 설정과 관리를 쉽게 그리고 누구나 알수 있고 할수 있게 하자는것이 주되는 목적이라고 할수 있다.

그러므로 Apache봉사기의 설정과 관리를 잘 하려면 httpd.conf파일의 문법과 사용규칙에 대하여 잘 알아야 한다.

제3장. 령역이름봉사기

국내망이든가 인터네트에 들어가보신 분들이라면 망에 얼마나 많은 콤퓨터들이 련결되여있는가를 보았을것이다. 그러면 이 많은 콤퓨터들이 어떻게 서로 상대방의 콤퓨터든가 말단들을 찾겠는가 하는 의문이 떠오른다.

뒤로 가면서 자세히 설명을 하겠지만 모든 콤퓨터들은 IP라는 10진수묶음을 리용하여 콤퓨터들을 찾는다. 그런데 사용자가 그 많은 콤퓨터들의 IP를 다 기억한다는것은 전혀 불가능한 일이다. 그렇다고 하여 전화번호를 리용하듯이 책에 의거하는것도 신통한 묘안은 못된다. 이것은 인터네트가 전 세계를 상대로 한다는 점을 놓고 볼 때 충분히 리해되는 문제이다.

《보다 쉽게 그리고 정확히 상대방의 콤퓨터를 찾을수 없겠는가?》

이러한 요구를 만족시켜주는 봉사기가 바로 령역이름봉사기이다.

령역이름봉사기는 그가 차지하는 중요성으로 하여 3대봉사기의 하나로 되고있다.

령역이름봉사기를 리용하면 지구상의 임의의 위치에서도 다른 곳에 있는 콤퓨터를 쉽게 찾을수 있다. 물론 일정한 규칙성을 따르는 조건에서이다.

이 장에서는 령역이름봉사기의 설치와 구축 및 리용방법에 대하여 설명한다.

이 장의 간단한 구성체계는 다음과 같다.

- 령역이름봉사기란 무엇인가?
- 령역이름봉사기의 설치와 구축
- 봉사기관리도구를 리용한 령역이름봉사기의 설정과 실행

제1절. 령역이름봉사기란 무엇인가?

령역이름은 DomainNameSystem의 략자로서 령역이름봉사기라고도 하는데 콤퓨터의 령역이름과 IP주소를 호상 변환하여 주는 역할을 한다. 여기서 령역이름봉사기에 대해 알려면 령역이름(domain name)이 무엇인가에 대하여 잘 알아야 한다.

령역이름이란 인터네트와 같은 대규모망의 콤퓨터나 사용자를 식별하기 위한 이름이라고 할수 있다. 령역이름은 나라, 조직의 종류, 조직의 명칭, 조직안에 설치되여 있는 콤퓨터 등을 령역이라고 부르는 론리적인 묶음들로 나누고 계층적으로 배렬하여 만든다.

례를 들어 IP주소가 192.168.8.59라는 콤퓨터를 찾아야 하는 경우를 보자. 이때 상대방의 IP는 잘 모르지만 사용자의 직무를 알고있는 경우가 있다. 다시말하여 192.168.8.59라는 IP를 사용하는 콤퓨터의 사용자가 조선민주주의인민공화국 교육성 교육정보쎈터 리철남이라는 사람이라고 할 때 192.168.8.59로 찾을수도 있지만 li.sec.edu.kr.으로도 찾을수 있다는것이다.

이때 li.sec.edu.kp.라는 이름을 리용하여 콤퓨터를 찾으려면 반드시 령역이름봉사기의 봉사를 받아야 한다.

령역이름봉사기는 의뢰기로부터 li.sec.edu.kp라는 콤퓨터를 찾는다는 요청이 들어오면 이 콤퓨터의 IP가 192.168.8.59라는것을 의뢰기에 알려준다. 또한 령역이름봉사기는 그 반대의 역할도 수행하여 준다. 다시말하여 요청에 따라 192.168.8.59라는 IP주소가 들어오면 li.sec.edu.kp로 바꾸어 주는 역할도 함께 수행하여 준다.

그러면 령역이름 즉 domain name과 IP는 어떤 관계가 있는가 하는 의문이 생긴다.

현재 인터네트를 비롯하여 많은 콤퓨터망들에서는 콤퓨터들 호상간의 통신을 TCP/IP규약을 리용하여 진행한다. 또한 콤퓨터들은 자기자신만이 유일하게 가지고 있는 IP로 구분된다.

콤퓨터에 대한 얼마간의 지식이 있는 분들은 IP에 대하여 알고 있으리라고 본다. 그런데 매개 콤퓨터에 대한 IP를 다 알고 있는다는것은 매우 어려운 일이다. 그러나 콤퓨터의 규약상 요구로 보나 설계요구로 보나 콤퓨터의 식별은 IP에 의하여 진행하여야 한다. 여기로부터 IP주소가 아닌 사람이 리용하기 쉽고 암기하기 쉬운 문자를 리용하면서도 실지 콤퓨터의 호출을 IP로 하여야 할 문제가 제기되였다.

이것을 해결한것이 령역이름봉사기이다.

령역이름봉사기는 령역이름에 해당하는 IP주소를 자료기지에서 찾아내여 리용한다. 결과를 놓고 말한다면 콤퓨터들은 령역이름으로 상대방의 콤퓨터를 찾는것이 아니라 IP주소를 사용하여 상대방 콤퓨터와 접속을 진행한다는것을 항상 명심하여야 한다.

만일 어떤 사용자가 웨브열람기의 주소창에 www.sec.edu라는 주소를 입력하였다고 하자. 이것은 사실 sec.edu라는 자료기지에서 www에 대한 주소레코드를 요청한것으로 볼수 있다. 그러면 이 레코드에 대한 요청은 LocalNameServer로 오게 된다.

만약 봉사기가 www.sec.edu라는 령역이름을 사용하는 사용자의 IP주소를 알고 있다면 응답을 진행한다. 그러나 그러한 주소에 대한 정보가 없고 sec.edu에 대해 취급하는 봉사기의 정보를 알고 있다면 해당 봉사기로 요청을 진행한다. 그러한 정보가 전혀 없다면 자기의 상위봉사기로 요청을 진행한다.

초기에 즉 인터네트와 같은 TCP/IP망이 나오기전까지는 주콤퓨터표를 사용하였다. 간단히 설명하면 조작체계안에 특정한 파일을 만들어 주고 거기에 콤퓨터의 이름과 IP주소의 대응표를 보관하여 그것을 리용하는 방법을 많이 리용하였다.

Linux조작체계에서는 etc/hosts파일에 령역이름과 IP주소에 대한 정보를 보관한후 IP와 일치하는 령역이름을 찾았다.

그림 3-1은 etc/hosts의 실례이다.

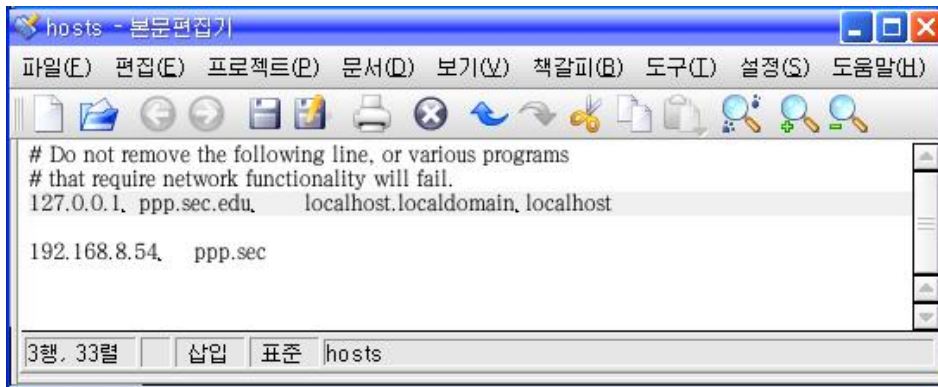



그림 3-1. etc/hosts파일의 실례

그림 3-1에서 볼수 있듯이 첫번째 행은 localhost라는 이름을 127.0.0.1이라는 주소에 할당한다. 콤퓨터망에서 127은 loopback망용으로 따로 예약된 특수한 주소이며 특히 127.0.0.1이라는 주소는 국부주기계용으로 예약된 Loopback주소이다.

Loopback주소란 국부콤퓨터가 원격콤퓨터에 주소를 붙이는 방법으로서 자기가 자신에게 주소를 할당할수 있도록 허용하는 규약이다.

그 다음 행은 ppp콤퓨터에 대한 정보이다. 이것은 192.168.8.54라는 주소에 ppp.sec라는 이름을 별명으로 할당한것이다.

이렇게 etc/hosts파일을 작성하면 ppp 또는 ppp.sec라는 이름을 가지고도 192.168.8.54라는 IP를 가지고 있는 콤퓨터를 찾을수 있다.

그러나 콤퓨터망이 커지면서 한개의 파일에 콤퓨터에 련결된 모든 콤퓨터에 대한 정보를 가지고 있어야 하는 우의 방식은 실용적이지 못한 면을 가지게 된다.

또한 오늘날의 인터네트를 생각해 볼 때 우의 방식으로는 짧은 시간동안에 수많은 콤퓨터들이 추가되고 삭제되는 상태에서 유지와 관리가 불가능하다고 볼수 있다.

그리하여 IP주소를 변환하는 봉사를 제공하기 위해 령역이름들에 대한 자료기지가 개발되여 그에 대한 정보를 봉사기에 보관하고 리용하기 시작하였다. 즉 매개 콤퓨터들이 접속하여야 할 콤퓨터의 정보를 각기 가지고 관리하는것 보다는 한대의 콤퓨터에 자료기지를 만들고 이것을 관리하며 다른 콤퓨터들은 이것을 리용하여 해당 콤퓨터를 찾는 방법을 리용하게 되였다. 즉 이름봉사를 매개의 망에 두어 요청이 들어올 경우 해당 요청에 대한 IP주소를 요청받은 이름봉사기가 검색한후 검색결과가 있다면 결과를 되돌려 주고 그렇지 않을 경우에는 정보를 가지고 있을만한 다른 이름봉사기에로 요청을 보낸다.

 령역이름(domain name)은 어떻게 만들어지는가.

령역(domain)은 콤퓨터소속기관의 종류를 나타내는 령역, 국가(지역)를 나타내는 령역, 령역안의 세부구분을 나타내는 부속령역 등이 계층적으로 구성된다. 국내콤퓨터망에서 널리 리용되는 <내나라>홈페지의 경우 kcc.co.kp라는 령역이름을 사용하고있다. 여기서 kp는 조선민주주의인민공화국을 의미하는 령역이름이며 co는 소속기관의 종류를 구분하는 령역이름이다. 따라서 령역이름을 보고 자기가 찾아야 할 콤퓨터가 있는 나라, 소속기관, 기업소명 등을 알수 있으며 보다 쉽게 상대방을 찾을수 있다.

이처럼 령역이름에 의하여 상대방을 찾을수 있는 봉사를 제공하여주는 봉사기가 바로 령역이름봉사기이다.

령역이름봉사기는 자기에게 요청한 봉사내용을 보고 자기가 관리하는 콤퓨터에 대한 정보이면 그에 대한 정보를 의뢰기에 넘겨주며 자기가 관리하지 않는 콤퓨터이면 자기에게 기록된 자료에 의하여 필요한 정보를 줄수 있는 다른 봉사기에게 요청을 진행한다. 그러한 정보도 없다면 자기의 상위봉사기에게 의뢰를 진행한다.

이러한 과정을 거쳐 령역이름봉사기는 의뢰기로부터 들어온 령역이름에 대한 해당한 정보(혹은 IP에 대한 정보)를 의뢰기에 알려준다.

이 과정이 령역이름봉사기의 동작과정이다.

그러면 령역이름봉사기의 설치와 구축에 대해 보도록 하자.

제2절. 령역이름봉사기의 설치와 구축

3.2.1. 령역이름봉사기의 설치

앞에서도 언급한것처럼 Linux조작체계에서의 설치에는 크게 RPM에 의한 설치와 원천파일로부터의 설치 두가지 방법이 있다.

- RPM으로 설치

령역이름봉사기를 설치하는데 필요한 RPM파일들은 다음과 같다.

표 3-1. 령역이름봉사기설치에 필요한 RPM파일

RPM파일	설명
Bind-9.2.4-2.RPM	령역이름봉사기에 필요한 패키지
Caching-nameserver-7.3-3.RPM	완충(caching)봉사기로서의 기능을 가지도록하기 위한 설정파일을 포함한 패키지

령역이름봉사기와 관련된 RPM파일들은 배포판CD에 포함되여있으므로 아래와 같이 설치하면 된다.

```
#rpm -Uvh bind-*
#rpm -Uvh caching-*
```

- 원천파일로 설치

령역이름봉사기를 운영하기 위해서는 봉사기측 데몬프로그람이 필요하게 된다. Linux조작체계에서는 BIND(Berkeley Internet Name Domain)라는 쏘프트웨어가 령역이름과 함께 운영된다. 이것들은 의뢰기-봉사기체계이다. 의뢰기는 resolver라고 하는데 resolver의 기능은 요청을 만들고 이 요청을 봉사기에 보내는 역할을 하는데 있다.

대부분의 UNIX체계에는 BIND가 이미 설치되여있다. 때문에 령역이름봉사기를 설치하기전에 /usr/sbin안에 in.named 혹은 named가 존재하는지 먼저 확인하여야 한다.

BIND에는 대표적으로 BIND 4와 BIND 8이 있다. 1998년 5월 11일 최종 판본 4.9.7이 개발되는것으로서 전통적인 BIND 4의 개발은 끝나게 되였으며 현재 BIND 8이 계속 개발되고있다. BIND 8은 그의 구성과 보안이 크게 개선되여 발표되였다. BIND 4와 BIND 8의 가장 큰 차이는 Boot파일의 변화이다. BIND 4의 경우는 named.boot이고 BIND 8은 named.conf로 바뀌였다. 물론 이름뿐만 아니라 그 문법도 바뀌였다. 기타 파일들은 거의나 같다.

BIND의 설치는 매우 간단하다. BIND원천파일을 구한 다음 압축을 풀고 생성된 등록부로 이동하여 지령을 주면 설치가 된다.

```
[root@ppp bind]# ls
Makefile bind-src.tar.gz src/
[root@ppp bind]# cd src/
[root@ppp src]# make clean depend all instal
```

설치를 다하였으면 아래의 그림 3-3과 같이 지령을 리용하여 정확히 설치되였는지 확인할수 있다.

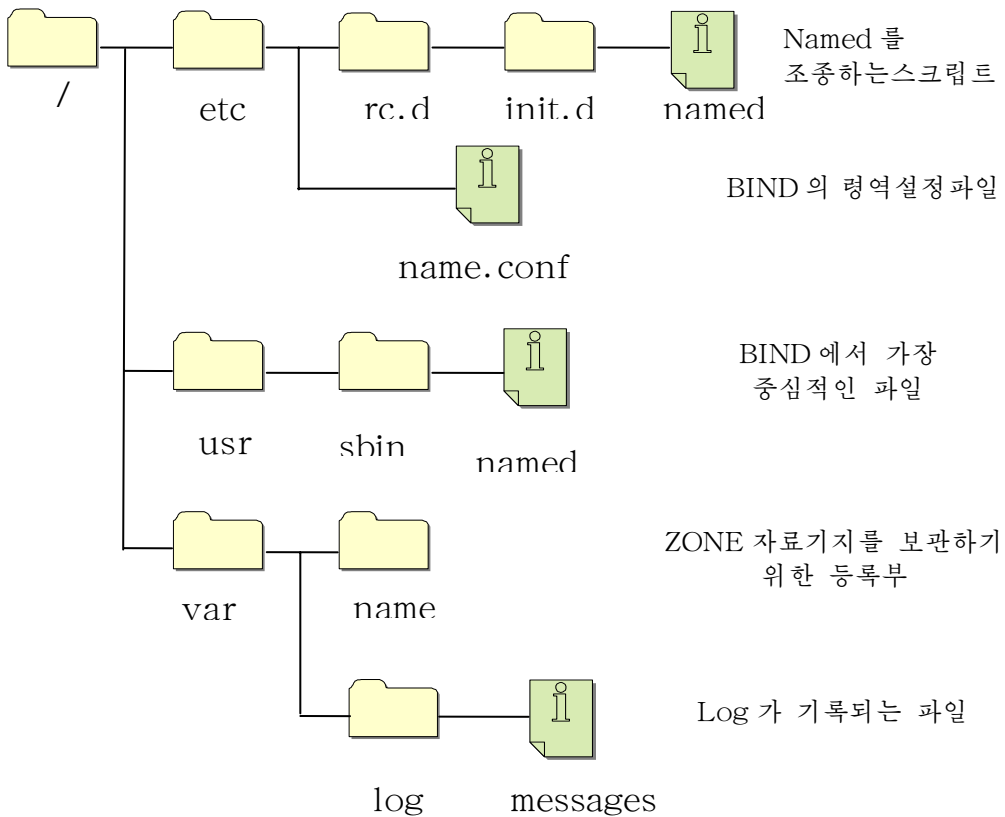



그림 3-2. BIND의 등록부구성

- 봉사기의 파일구성

봉사기의 설치가 다 되였으면 해당한 파일들의 설치가 끝났다고 볼수 있다.
그림 3-2에서는 BIND의 등록부구성을 보여주었다.

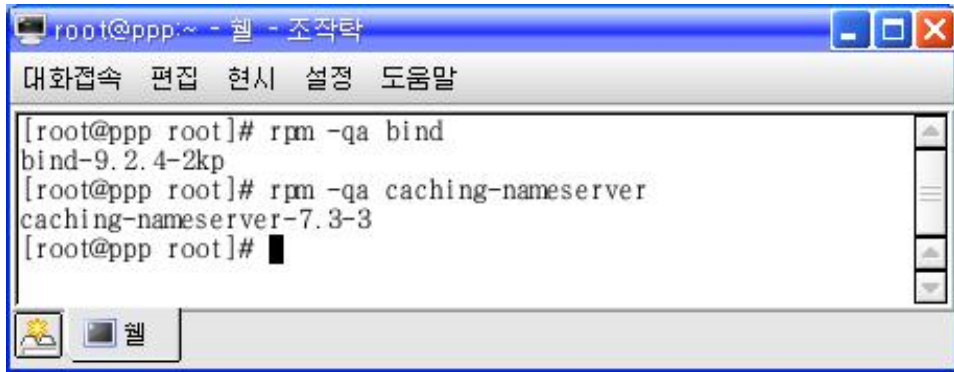



그림 3-3. DNS봉사기 설치의 확인

3.2.2. 구성파일의 설정

의여기서는 봉사기구축의 핵심으로 되는 구성파일들의 설정에 대하여 취급한다. 여기서 한가지 명심하여야 할 점은 BIND설정파일들의 발전속도가 빠르므로 판본들사이의 파일의 설정에서 약간의 차이가 있다. 그러므로 파일들의 설정에 대한 설명을 진행함에 있어서 이전 판본들과의 공통적인 문제들에 기본을 두고 설명을 진행한다. 그러므로 설명을 진행할 때 그림과 약간 차이나는 부분이 있으면 판본들 차이점이라는데 류의하면서 리해를 하여야 한다.

- /etc/host.conf파일

host.conf파일은 resolver추가설정항목들을 보관하는 파일이다. 이 파일에서 주석은 #로 시작한다. 이 파일에 제시된 추가설정항목들은 resolver에게 어떤 봉사들을 어떤 순서로 사용할것인지 알려준다. 이 파일에서는 순서가 중요하다. resolver는 제시된 추가설정의 첫 항목으로부터 시작하여 아래로 가면서 순서대로 수행된다.

표 3-2. host.conf파일의 추가설정과 그의 기능

추가설정	기 능
order	①host: /etc/host파일에서 이름을 검색한다. ②bind: 령역이름이름봉사기에게 주소를 요구한다. ③nis: Network Information Service규약을 사용하여 주소를 알아낸다.
Alert	체계에 접근하려고 시도하고있는 원격체계의 주소를 검사한다. On과 off추가설정항목을 사용하여 동작과 취소를 수행한다.
Nospoof	체계에 접근하려고 시도하고있는 체계의 주소를 확인한다.
Trim	Local host파일들을 검색하기 위해 령역이름을 없애고 주콤퓨터이름만을 검사한다. 이 추가설정항목은 주콤퓨터이름만을 사용하여 IP주소를 검색한다.
Multy	Local host파일을 검색하기 위해 주콤퓨터가 여러개의 IP를 가지고 있도록 한다. On과 off추가설정항목을 사용하여 동작과 취소를 진행한다.

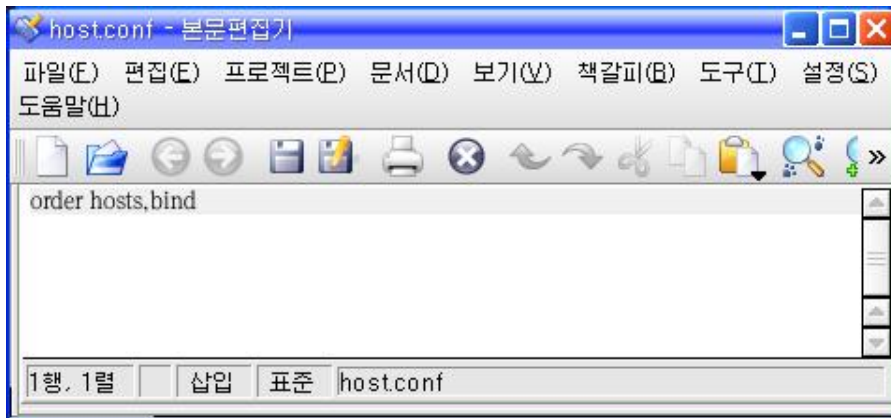



그림 3-4. host.conf파일의 실례

표 3-3. resolv.conf파일의 열쇠단어와 설명

열쇠단어	설명
Domain	자기콤퓨터의 령역이름을 현시한다.
nameserver	자기콤퓨터가 접근할수 있는 이름봉사기의 IP주소를 현시한다. 보통 search에 명시되는 령역의 IP주소이다. 최대 3개까지 등록이 가능하다.
Search	주콤퓨터의 이름을 봉사기로 보내기전에 확장하는데 사용되는 령역의 목록을 제시한다. 목록은 최대 6개까지 등록가능하다.

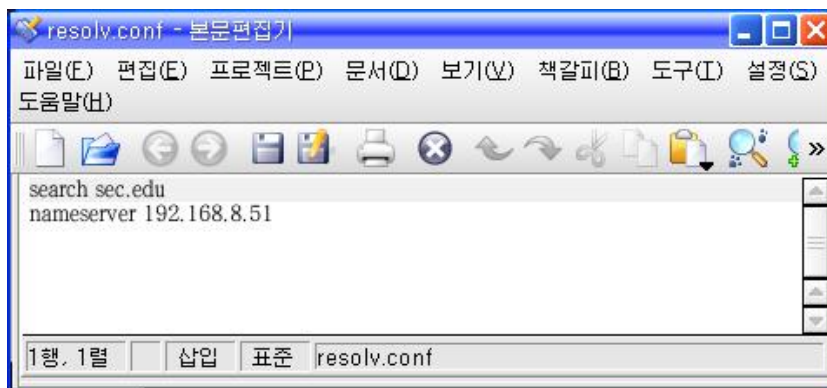



그림 3-5. /etc/resolve.conf파일의 실례

- /etc/resolv.conf

resolv.conf파일은 resolver가 체계에 접근할수 있도록 령역이름봉사기들의 주소를 제공한다. 이 파일에는 표3-3에 서술한것과 같은 열쇠단어들이 있다. 우의 표 3-3에 열쇠단어와 그에 대한 설명을 주었다.

- /etc/named.boot

이 파일은 이름봉사기파일에서 기본설정파일들중의 하나이다. /etc/named.boot파일은 BIND4가 기동할 때 이름봉사기에 읽혀지며 설정되여있는대로 다른 파일들을 포함하거나 동작을 수행한다. 정확히 알아두어야 할 점은 이 파일이 BIND 4에서 리용되는 파일이라는것이다. BIND 8에서는 이 파일대신에 name.conf라는 파일을 리용한다. named.boot파일은 named.conf파일과 같은 역할을 하는 파일로서 BIND4판본에서 사용되여 왔다. 두 파일사이에는 약간의 문법차이가 있으므로 두 파일에 대한 설정에 대하여 모두 보기로 하자.

이 파일에서 기본적으로 알아야 할것은 directory, cache, primary 세가지이다.

```
Directory  /var/named
```

우의 부분은 정보를 가지고있는 이름봉사기의 자료파일들이 위치하고있는 등록부를 지정하는 부분이다. 즉 다른 추가선택항목들에서 지정하고있는 파일들이 여기에서 지정한 등록부에 위치하게 된다는것이다.

```
Cache.named.ca
```

우의 부분은 root이름봉사정보를 가지고있는 자료기지파일을 지정한다. BIND는 다른 령역정보를 root이름봉사기로부터 얻는다. 이때 root이름봉사기의 정보를 이 파일로부터 얻게 된다. 우의 설정에서 보면 named.ca파일은 /var/named/named.ca에 위치하고 있다. 여기서 주의를 돌려야 할것은 cache다음에 있는 《.》이다. 보통 우리가 령역이름을 적을 때 www.sec.edu라고 적는다. 그러나 마지막에 《.》을 적은 www.sec.edu.이 정확한 이름이다. 다시말하여 최상위 령역은 《.》이 되는것이다. 때문에 이름봉사기가 정보를 찾기 위하여 가장 먼저 질문을 보내는곳도 《.》을 관리하는 이름봉사기가 되는것이다.

```
primary          sec.edu        sec.zone
```

우의 부분은 해당한 령역에 대하여 primary이름봉사기로 동작한다는것을 의미한다. 다시말하여 sec.edu에 대하여 이 체계가 primary이름봉사기로 동작하며 자료기지파일은 /var/name/sec.zone파일이라는것을 나타낸다.

Secondary이름봉사기로 동작할 경우에는 primary를 secondary로 바꾸어주면 된다.

```
primary          8.168.192.in-addr.arpa     sec.rev
```

우의 부분도 primary이름봉사기로 동작한다는것을 의미하는 부분이다. 다시말하여 192.168.8.에 대해 이 체계가 primary이름봉사기로 동작하며 자료기지파일은

/var/name/sec.rev파일이라는것을 나타낸다.

뒤에서 자세히 설명을 진행하겠지만 우의 두 파일 sec.rev와 sec.zone는 모두 이름봉사기의 자료기지파일들이다. 그렇다면 왜 같은 이름을 달고있는 파일이 두개 필요한가 하는 의문이 생긴다. 그것은 앞에서도 설명했지만 령역이름의 역할이 령역(Domain)을 IP로 바꾸어주는 역할과 함께 IP를 령역으로 바꾸어주는 역할도 하기때문이다. 즉 령역을 IP로 바꾸어주는 역할을 할 때에는 sec.zone의 정보를 사용하며 IP를 령역으로 바꾸어주는 역할을 할 때에는 sec.rev의 정보를 사용한다.

- /etc/named.conf

named.conf파일은 앞에서 설명한 named.boot와 같은 역할을 한다. BIND 8판본에서부터 령역이름기동파일로 쓰이기 시작하였다. 이것이 BIND4와 BIND8의 가장 큰 차이이기도 하다. 기본적인 구성은 named.boot파일과 비슷하지만 많은 부분이 추가확장되였다. 또한 BIND4와의 혼동을 막기 위해서 파일이름도 변화시켰다. 이제 설명하는 named.conf파일은 앞에서 설정한 named.boot파일을 BIND8의 문법에 맞게 변환한것이다.

그림 3-6은 named.conf의 실례파일이다.

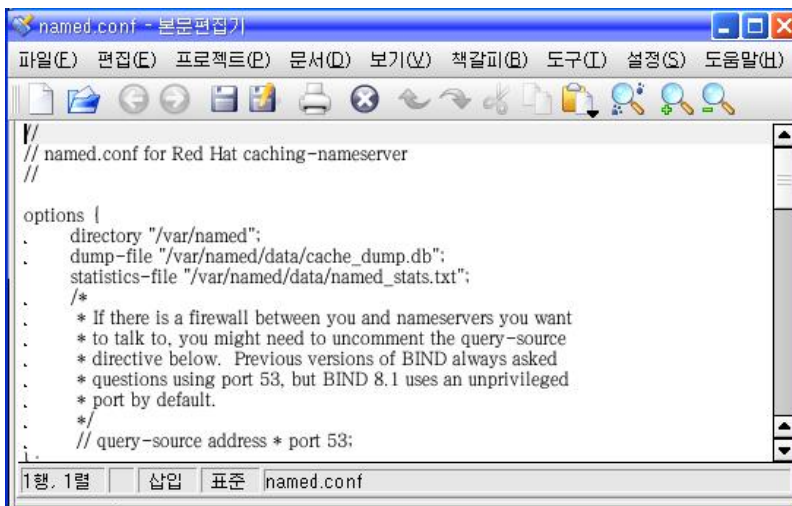



그림 3-6. named.conf의 실례

그러면 파일의 내용을 자세히 보도록 하자.

```
option{
   directory “/var/named/”;
};
```

우의 부분은 봉사기가 참조할 파일들이 있는 등록부를 설정한다. 정확히 말하면 앞에서

설명한 전역구성과 표준을 정의한 파일들이 있는 등록부를 설정한다. Directory열쇠단어는 named가 읽고 쓸 등록부의 위치가 /var/named임을 정의하고 있다. 여기서 주의할 점은 options와 directory가 모두 마지막에 《;》으로 끝난다는것이다. 편집과정에 이 부분에 주의를 돌려 오유가 발생하지 않도록 하여야 한다.

```
zone  "0.0.127.in-addr.arpa" {
   type master;
   file  "0.0.127.in-addr.arpa.zone";
};
zone  "8.168.192.in-addr.arpa" {
   type master;
   file  "192.168.8.rev";
};
```

우의 부분은 127.0.0령역과 192.168.8 령역의 콤퓨터들을 관리하기 위한 설정이다. 자세한 설명은 앞에서 언급하였으므로 략한다.

```
zone  "localhost" {
   type master;
   file  "localhost.zone";
};
zone  "sec" {
   type master;
   file  "sec.zone";
};
```

우의 부분은 sec.edu이라는 령역의 주콤퓨터들을 관리하기 위한 설정이다.

- 지역령역(ZONE)의 설정

여기에는 령역이름이 관리할 령역에 대한 정보를 가지고있는 파일들에 대한 설정이 포함된다. 먼저 파일들에 대한 설명을 진행하기전에 령역이름이 어떠한 과정을 통해서 어떻게 동작을 수행하는가에 대해서 먼저 간단히 보도록 하자.

- 령역이름의 동작

령역이름봉사기의 동작과정을 제대로 리해하는것은 령역이름봉사기를 안정하고 정확히 운영하는데서 중요한 문제로 제기된다.

령역이름의 동작을 리해하기 위해서는 앞에서 설명한 named.conf파일에 대해서 잘 리해해야 한다.

령역이름은 《.》으로 표시되는 root라고 하는 최상위 계층을 가지고 아래로는 앞에서도 본것처럼 주소의 마지막에 붙는 com, edu, net, org, mir, gov, uucp 등과 같은 최상위

령역(TLDs: Top Level Domains)으로 이루진 계층적인 구조를 가지는 체계이다.

그러므로 특정한 이름을 가진 콤퓨터를 찾을 경우 그 요청은 최상위 계층인 《.》에서부터 시작하여 그 아래에 있는 계층을 차례로 찾아 내려가게 된다.

례를 들어 www.sec.edu라는 주소를 가진 콤퓨터를 찾을 경우 이름봉사기는 우선 요청된 령역이름에서 edu라는 최상위 령역을 찾아보고 edu라는 령역을 관할하는 이름봉사기를 찾는다. 다시말하여 《.》봉사기에 요청을 하게 되며 《.》봉사기는 다시 edu령역을 관할하는 이름봉사기들에 대한 정보를 목록으로 넘겨준다. 그러면 이 이름봉사기의 정보에 기초하여 sec.edu라는 령역을 관할하는 이름봉사기를 찾게되며 찾아낸 봉사기로부터 www.sec.edu라는 콤퓨터의 정보를 얻어내게 된다. 다음 얻어낸 정보를 의뢰기에 알려주며 의뢰기는 이 정보에 기초하여 www.sec.edu라는 콤퓨터를 찾게 된다.

- zone 파일설정

파일의 설정에 앞서 주콤퓨터이름을 정할 때 주의해야 할 점에 대해서 간단히 보도록 하자.

무엇보다도 건반우에 있는 모든 문자를 주콤퓨터이름으로 사용할수는 없다. 사용할수 있는 문자는 우선 a - z 까지의 문자와 수자(0-9) 그리고 《-》문자만을 사용할수 있다.

다음으로 주의해야 할 점은 주콤퓨터이름은 대문자와 소문자를 구별하지 않는다는것이다. 실제로 www.sec.edu라고 요청을 하는것이나 www.SEC.EDU로 요청을 하는것이나 모두 같은 의미를 가지며 따라서 결과는 같다.

우와 같은 문제들에 주의를 돌리면서 콤퓨터의 이름을 정하여야 한다.

표 3-4는 zone파일에서 사용하는 정보의 형(type)들이다.

표 3-4. zone에서 사용하는 type와 그의 기능

Type	기능
A	주콤퓨터의 인터네트주소.(표준값)
CNAME	대상주콤퓨터의 이름이 별명일 경우 별명의 원래 이름
HINFO	대상주콤퓨터의 CPU와 조작체계형태
MINFO	대상주콤퓨터의 전자우편정보
MX	대상주콤퓨터의 우편봉사기
NS	대상주콤퓨터의 이름봉사기
PTR	대상주콤퓨터의 IP주소가 입력될 경우에는 주콤퓨터이름을 나타내고 그 밖의 경우에는 정보들에 대한 point정보
SOA	대상주콤퓨터의 Start of Arthority
TXT	대상주콤퓨터의 Text정보
UINFO	대상주콤퓨터의 사용자정보
WKS	대상주콤퓨터가 지원하는 주요봉사들의 정보

Zone파일에는 두가지가 있다. 하나는 령역이름을 IP로 변환할 때 사용하는 파일이며 다른 하나는 IP를 령역이름으로 변환할 때 사용되는 파일이다.

먼저 령역이름을 IP로 변환하는 파일에 대해 보기로 하자.

- /var/named/sec.zone

여기서 강조할 문제는 sec.zone이라는 파일이름이 절대적인 파일이름이 아니라는것이다. 만약 linuxbible.com이라는 령역에 대해서 설정을 할 경우에는 linuxbible.zone라는 이름으로 설정되며 여기에 맞게 앞에서 설명한 /etc/named.conf(BIND 4인 경우에는 name.boot)를 적절하게 설정하면 된다.

그림 3-7. /var/named/sec.zone의 실례

앞에서 설명한 바와 같이 name.conf파일에서는 sec.zone파일을 sec.edu령역을 설정하는 파일로 설정을 하였다. 여기에서 명백히 알아두어야 할 점은 sec.zone파일은 령역이름을 IP주소로 바꾸어 줄 때 필요한 정보를 가지고 있는 파일이라는것이다. 그러면 sec.zone파일에 대하여 보기로 하자.

우의 그림 2-7은 /var/named/sec.zone의 실례파일이다.

```
@  IN    SOA   192.168.8.54.   root.localhost (
```

우에서 제일 처음에 있는 부분은 SOA레코드라고 하는데 zone파일은 항상 이 SOA레코드로부터 시작한다. @는 실지 콤퓨터의 이름을 나타낸다. 여기서는 192.168.8.54를 나타내는것이다.

IN(internet)은 클라스명이다. HS, HESIOD, CHAOS와 같은 클라스도 있지만 일반적으로 사용되지는 않는다. Zone파일에서 《@》는 앞에서 설명한것처럼 실지 콤퓨터의 이름을 나타내기 때문에 《.》를 사용한다.

```
3 ; serial
```

Serial은 Secondary가 zone 파일을 수정하였는지를 알아보기 위해 존재한다.

일반적으로 Secondary봉사기는 주기적으로 Primary봉사기의 zone파일의 SOA레코드를 검사한다. 다시말하여 zone파일의 내용이 변화되였는가를 검사하는것이다. 즉 변화되였는지 안되였는지의 여부를 확인하려면 이 serial번호를 확인하면 된다. Primary봉사기의 serial번호가 Secondary봉사기가 가지고 있는 serial번호보다 클 경우 Secondary봉사기는 Primary봉사기의 zone파일이 변화되였다고 판단하며 Primary봉사기의 zone파일을 전송해온다. 만약 Primary봉사기의 serial번호가 작거나 같을 경우에는 전송을 생략한다. 그러므로 령역이 갱신되거나 변경될 경우에는 serial번호를 증가시켜주는것이 Secondary봉사기와 Primary봉사기의 동기화를 위해서 필요하다.

보통 serial번호의 갱신은 임의의 수자를 쓰기보다는 가장 최근에 수정한 날자를 《날자(YYYYMMDD)+두등록자리》수자의 형식으로 취하는것이 좋다.

```
21600 ; refresh
```

우에서 실례로 보여준 refresh번호는 Secondary에서 Primary의 zone자료기지파일의 수정여부를 검사하는 시간주기를 나타낸다. Refresh에 사용되는 수자는 주기를 초로 환산한것이다. 그러므로 우와 같이 설정하였다면 하루에 4번 Secondary봉사기가 Primary봉사기의 zone파일의 갱신여부를 검사하게 된다. 령역에 대한 갱신을 자주 하여야 할 경우에는 10800정도로 설정하는것이 좋다.

주기가 짧을수록 Secondary봉사기와 Primary봉사기의 동기화가 보다 긴밀히 이루어진다. 하지만 Primary봉사기가 령역이름 NOTYFY통보문을 보내면 Secondary봉사기가 갱신이 진행된것으로 알고 Primary봉사기의 zone파일을 검사하기때문에 이 값을 무조건 짧게 설정할 필요는 없다.

```
7200 ; retry
```

Retry는 Secondary봉사기가 Primary봉사기의 zone파일에 대한 갱신여부를 확인할 때 만약 Primary봉사기가 그에 대한 응답이 없을 경우 Secondary봉사기가 재확인을 요구하는 주기를 나타낸다. 일반적인 경우에는 1800으로 설정하여 30분을 주기로 다시 응답을 요구하도록 설정한다. Retry에 사용되는 수자는 주기값을 초로 환산한것이다.

```
604800 ; expire
```

Expire는 Secondary봉사기가 Primary봉사기에 재응답을 요청할수 있는 시간제한

을 나타낸다. Expire에 사용되는 수자도 주기를 초로 환산한것이다. 만약 expire에 설정된 시간동안 Primary봉사기로부터 응답을 받지 못했을 경우 Secondary봉사기는 이전의 여벌자료에 대해서 더 이상 유효성을 인정하지 않는다.

보통 이 값에는 시간적으로 여유를 많이 준다. 그것은 Secondary봉사기의 목적의 하나가 Primary봉사기에 대한 여벌을 제공하는데 있기 때문이다.

```
86400 ; ttl
```

우에서 서술한 ttl은 Minimum으로 표기하기도 한다. ttl은 지정된 령역의 표준생존기간 (TTL:Time To Live)을 초로 환산한것이다. 즉 다른 이름봉사기가 zone파일을 가지고 갔을 때 그 자료에 대한 유효기간이라고 리해하면 된다. 우의 실례에서는 86400으로서 표준 TTL을 하루 즉 24시간으로 설정하였다. 보통 24시간으로 설정한다. TTL값은 령역에 대한 갱신이 자주 일어날 경우에는 짧게 설정하는것이 좋고 그렇지 않고 안정적일 경우에는 길게 설정하는것이 좋다.

지금까지 SOA레코드의 구성 요소들에 대해서 보았다. 모든 자료구성요소들은 전체 령역에 영향을 미친다. 이 항목들중에는 원격봉사기에 영향을 미치는것도 있다. 그러므로 령역관리자는 령역에 알맞는 설정을 진행하여야 한다. 그것은 이 파일들의 설정과 관리가 전적으로 령역관리자에 의하여 진행되기 때문이다.

```
@  IN      NS      192.168.8.54
```

우의 부분은 해당 령역의 이름봉사기를 정의한것이다. 즉 sec.edu를 설정한것이라고 볼수 있다. 이름봉사기는 콤퓨터망의 특징에 따라 여러개를 설정할수도 있다. 우의 설정실례와 같이 NS 레코드에 이름마당을 빈 공간으로 사용할수 있다. 이 마당에 나타난 값이 named.conf파일에 정의된 sec.edu령역을 나타내는 @이였다. 그러므로 NS레코드에 정의된 령역은 sec.edu령역의 이름봉사기를 정의한것이다.

```
kim       IN      MX     1       192.168.8.54
```

MX(Mail eXchange)레코드는 령역의 전자우편봉사기에 대한 설정이다. 보통 대형봉사기들은 웨브봉사기, 전자우편봉사기 등을 분리하여 운영을 진행한다. 우의 실례와 같이 설정이 되여있는 상태에서 sec.edu의 testuser라는 등록자리사용자에게 전자우편을 보낸다고 하자. 전자우편을 리용하여 본 사람이라면 알수있듯이 보통 우리는 다음과 같이 전자우편주소를 적는다.

```
testuser@sec.edu
```

우와 같은 주소로 보낸 전자우편은 우의 설정으로부터 sec.edu로 보내지게 된다. MX뒤에 있는 수자는 전자우편봉사기의 우선순위를 정해주는 Priority_Number이다. 이

Priority_Number는 작을수록 우선 순위가 높다.

MX의 설정시 주의해야 할 점은 MX레코드를 CNAME으로 설정된 령역이름으로 설정하지 말아야 한다는것이다. 그것은 이러한 경우에 몇개의 MTA(Mail Transfer Agent)는 우편을 전송할 경로를 찾지 못하여 우편을 전송하지 못하는 경우가 생기기 때문이다.

```
ppp.sec.        IN    A      192.168.8.55
```

우와 같은 설정은 sec.edu령역에 주콤퓨터를 설정한것이다. 우의 설정은 192.168.8.55라는 IP주소에 ppp.sec.edu라는 령역이름을 할당했다는것을 의미한다. 령역을 설정하는 방법은 우와 같이 IP주소에 령역이름을 지정하는 방법과 별명(alias)을 지정하는 2가지 방법이 있다.

```
kim     IN     A      192.168.8.55
mmm  IN CNAME  kim
```

우의 설정에서도 192.168.8.55라는 IP주소에 kim.sec.edu라는 령역이름을 할당한것과 같은 결과를 나타낸다. 그리고 다음 행에서는 이 령역이름에 다시 mmm이라는 별명을 설정하였다. 즉 mmm으로 접속을 하거나 혹은 kim으로 접속을 하면 모두 192.168.8.55라는 주소로 접속이 진행된다. 이러한 별명은 하나의 IP주소에 여러개의 령역이름을 부여할 경우에 많이 사용된다.

그러나 하나의 IP주소에 여러개의 령역이름을 부여할 경우 꼭 별명을 써야하는것은 아니다.

sec.zone파일의 마지막에 다음과 같이 추가를 하면 192.168.8.54라는 IP주소에 server라는 령역과 mail이라는 2개의 령역이름이 할당되게 된다.

```
server    IN     A      192.168.8.54
mail      IN     A      192.168.8.54
```

이와 같은 방법으로도 여러개의 령역이름을 부여할수 있다.

지금까지 령역이름을 IP주소로 바꾸어줄 때 필요한 sec.zone파일에 대해서 보았다. 다음으로 IP주소를 령역이름으로 바꾸어줄때 사용하는 192.168.8.rev파일에 대해 보기로 하자.

- /var/named/192.168.8.rev

sec.zone파일과 마찬가지로 192.168.8.rev파일의 이름도 절대적인 파일이름은 아니다. 자기의 환경에 맞게 이름봉사기관리자가 임의로 파일이름을 정하고 이에 맞게 앞에서 설명한 /etc/named.boot와 /etc/named.conf파일을 적절하게 설정하면 된다.

령역이름의 역할은 앞에서 설명한것처럼 령역이름을 IP주소로 바꾸어주는 역할과 IP주소를 령역이름으로 바꾸어주는 2가지 역할이 있다.

여기에서 설명하게 되는 일명 역령역파일이라고 하는 192.168.8.rev파일은 IP주소

를 령역이름으로 바꾸어줄 때 필요한 정보를 가지고있는 파일이다. 이 개념을 잘 기억하여야 앞에서 진행한 설명과 혼돈을 가져오지 않는다. 앞에서 설명한 sec.zone파일과 반대되는 사명을 수행하므로 이 파일을 역령역파일이라고 부른다.

192.168.8.rev파일을 역령역파일이라고 정의하는 또다른 리유가 있다. 그것은 192.168.8.rev에는 IP주소가 거꾸로 기록되여있기때문이다. 즉 192.168.8.55라는 IP주소가 있다면 이 IP주소는 55.8.168.192.in-add.arpa와 같은 형식으로 기록되여있다. 령역이름과 대칭시킬 경우에는 다시 이 주소를 거꾸로 해석한다. 그림 3-8은 192.168.8.rev의 실레파일이다.

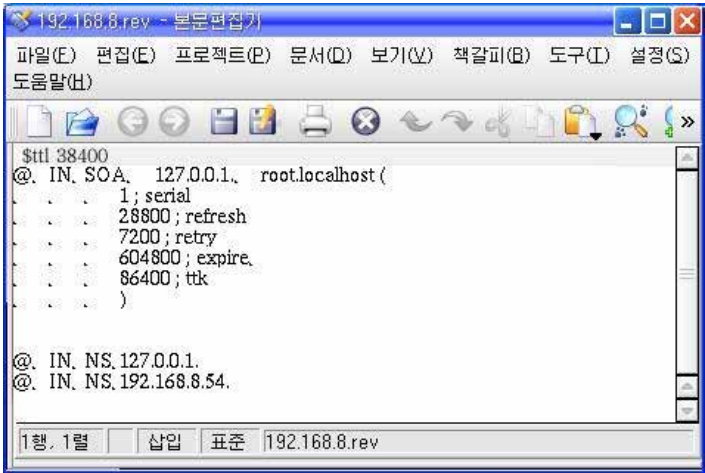



그림 3-8. 192.168.8.rev의 실레

그러면 192.168.8.rev파일의 구체적인 설정과 그의 역할에 대해 보기로 하자.

그림 3-8에 있는것처럼 192.168.8.rev파일도 sec.zone파일과 같이 SOA레코드로 시작을 한다. 또한 NS레코드령역도 가지고 있다. 이들에 대한 설명은 이미 sec.zone파일에 서 진행하였으며 기능도 서로 같으므로 여기에 대한 설명은 생략하기로 한다.

```
primary         8.168.192.in-addr.arpa     sec.zone
```

우의 설정은 이름봉사기가 192.168.8.x의 IP주소 모두를 관리한다라는 의미이다. 그러므로 IP주소의 마지막 부분만을 써주게 되면 이름봉사기는 마지막 IP주소앞에 192.168.8을 붙여 IP주소를 해석하게 된다.

```
55.    IN.    PTR.  kim.sec.
57.    IN.    PTR.  won.sec.
```

우와 같이 설정을 진행하게 되는데 이때 55는 완전한 주소가 아니기 때문에 현재 설정된 령역으로 해석을 진행하여 55.8.168.192.in-addr.arpa를 령역이름으로 받게 된다. 즉 kim.sec.edu라는 령역이름에 192.168.8.55라는 IP주소를 되돌려주게 된다.

여기에 대해서는 앞에서 설명한 /etc/named.conf파일에 대해 제대로 리해했다면 쉽게 알수 있다.

192.168.8.rev파일설정에서 주의해야 할 점은 IP주소에 넘기기되는 모든 령역이름의 뒤에 《.》이 붙어있다는것이다. 그것은 BIND가 《.》으로 끝나지 않는 문자렬은 주콤퓨터이름으로 해석하기 때문에 만약 앞에서와 같은 설정에서 《.》을 붙이지 않는다면 이름봉사기는 kim.sec.edu를 kim.sec.edu.sec.edu로 잘못 해석하게 된다.

2.2.3. 완충전용이름봉사기(caching only name server)의 설정

이름봉사기의 역할은 크게 2가지로 나눌수 있다. 요청이 들어왔을 때 root 봉사기와 같은 다른 봉사기들로부터 요청에 대한 정보를 얻어오는것이 첫번째 역할이며 다음으로 자신이 가지고있는 정보로 요청에 대한 응답을 하는것이 두번째 역할이다. 여기서 완충전용이름봉사기(caching only name server)는 후자에 속한다. 이름봉사기는 완충(caching)을 진행한다. 또한 완충을 하지 않는 봉사기는 없다. 그러므로 완충전용이름봉사기의 설정은 모든 봉사기에 관한 문제라고 말할수 있다. 완충전용이름봉사기의 기능은 한번 요청한 정보를 봉사기가 기억해두었다가 다시 동일한 요청이 왔을 때 기억해두었던 정보를 가지고 바로 요청에 응답을 할수 있게 하는것이다. 이 기능을 리용하면 봉사의 질과 속도를 개선하는데서 큰 문제를 해결할수 있다.

- /var/named/named.ca

이 파일은 hints파일이라고도 하는데 named가 고속완충기를 초기화할 때 사용하는 정보를 보관하고있는 파일이다. 이 파일에는 root 봉사기들에 대한 이름과 주소가 보관되여있다. 즉 이 파일은 named가 기동될 때 봉사기가 root봉사기를 찾도록 도와주는 역할을 한다. 그리고 이 파일은 named가 시작될때에만 읽어들여지며 named가 다시 재시동할 때까지는 읽어들이지 않는다. 이 파일의 이름은 절대적인것이 아니다. 가장 일반적으로 사용되는 파일이름으로는 named.ca, named.root 등이 있다.

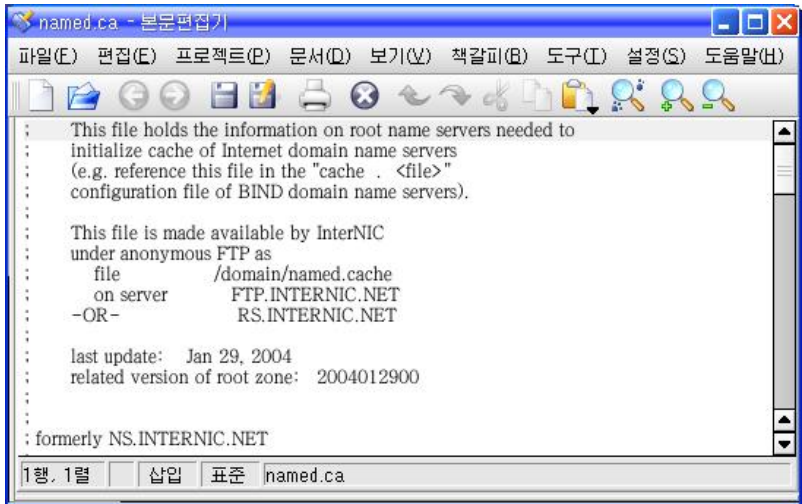



그림 3-9. named.ca파일의 실례

파일 앞부분의 《;》으로 주석처리된곳에서는 named.ca파일에 대한 간단한 설명과 이 파일을 얻을수 있는 FTP주소에 대해서 설명하고있다.

이 파일은 이름봉사기와 그 이름봉사기의 IP주소만으로 구성되여있다.

이 파일에는 A.ROOT-SERVERS.NET에서 M.ROOT-SERVERS.NET까지 root 이름정보들이 보관되여있다.

완충전용이름봉사기(caching only name server)설정에 쓰이는 named.ca파일이나 named.local파일은 보통 체계를 설치할 때 자동적으로 생성되며 named.ca파일을 주기적으로 갱신해준다. 또한 named.local파일은 어떤 체계에서나 모두 같은 모습을 하고있다.

아래의 그림 3-10에 name.local파일의 실례를 보여주었다.

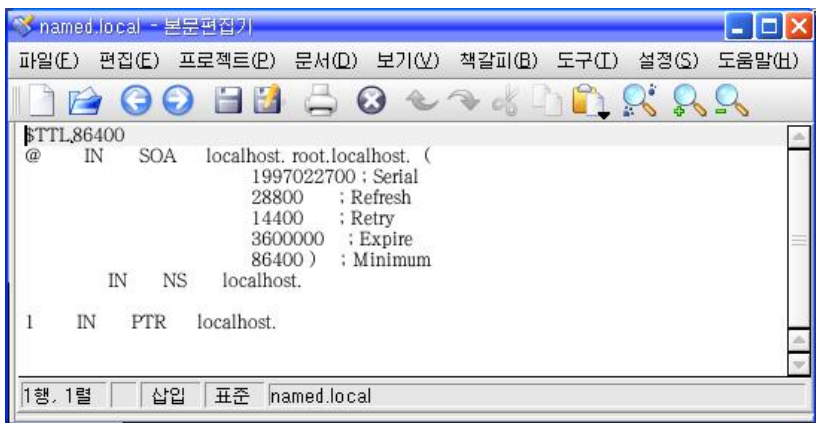



그림 3-10. name.local파일의 실례

제3절. 봉사기관리도구를 리용한 령역이름봉사기의 설정과 실행

앞에서 우리는 령역이름봉사기의 설치를 파일들을 수정하거나 설정하여 진행하였다. 물론 이 방법이 기초적이면서 가장 정확하며 또 봉사기의 관리에서 실수가 없도록 하는 방법인것은 사실이다. 그러나 처음 봉사기를 구축하는 사람들에게 매우 어렵고 하기 힘든 방법이 바로 파일들을 직접 변화시켜가면서 봉사기의 설정과 관리를 진행하는 과정으로 된다는것은 누구나 잘 알것이다. 앞의 과정을 거쳐서 봉사기의 설정을 해보면서도 느낀바이겠지만 파일들을 직접 조종하는것은 힘든 일이다. 이러한 문제를 쉽게 해결해주는것이 앞에서도 언급된 봉사기관리도구이다. 곰곰히 생각해보면 앞에서 봉사기관리도구에 대해서 볼 때 봉사기항목에서 령역이름봉사기를 본 생각이 날것이다.

여기서는 앞에서 배운 령역이름봉사기의 설정에 기초하여 봉사기관리도구를 리용하여 봉사기를 구축하고 설정하는 방법과 이 도구를 리용한 령역이름봉사기의 실행과 정지에 대해 취급한다.

3.3.1. 봉사기관리도구를 리용한 령역이름봉사기의 설정과 관리

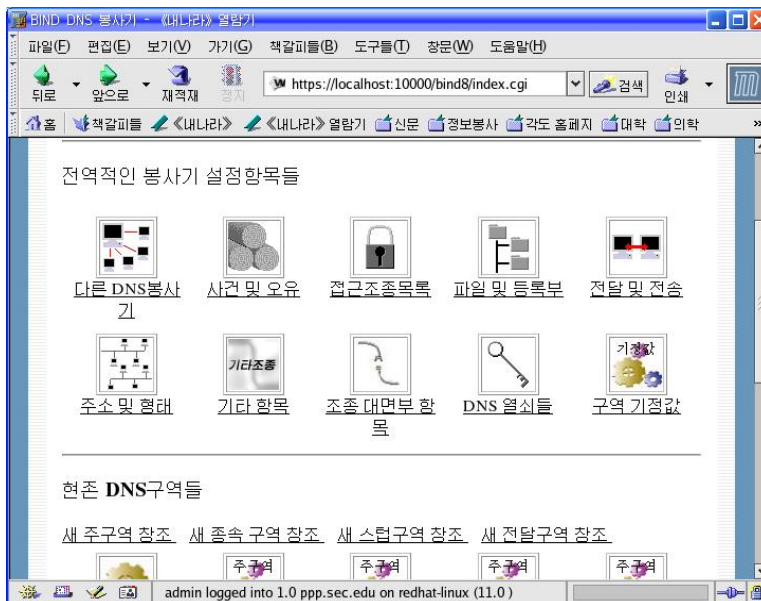



그림 3-11. BIND DNS봉사기관리도구항목의 대면부

봉사기관리도구의 실행에 대해서는 이미 앞에서 언급되였으므로 다시 반복하지는 않는다. 관리도구를 실행시키면 그림 2-7과 같이 관리도구가 실행된다. 여기서 《BIND 령

역이름봉사기》선택항목을 찰칵하면 령역이름봉사기의 설정과 관리를 진행하는 창문이 펼쳐지게 된다.(그림 3-11)

그림 3-11의 대면부에 나오는 선택항목들을 보면 어떤 내용들에 대한것을 관리, 설정하는 항목들인지 인차 알수 있다. 관리도구의 리용은 쉽게 할수 있으므로 여기서는 이것을 리용하는 방법에 대해서 모두 설명하지 않고 이름봉사기의 형성과 관리에 대해서만 보기로 하자.

먼저 새로운 주구역을 창조하여야 한다.

《새로운 주구역 창조》항목을 찰칵하면 새로운 주구역을 창조하는데 필요한 대면부가 나타난다.(그림 3-12)

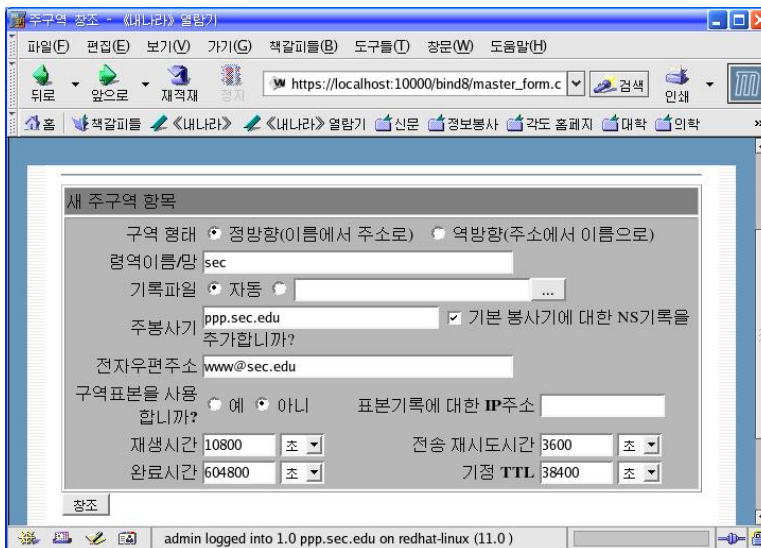



그림 3-12. 새로운 주구역 창조 대면부

새로운 주구역창조에 필요한 값들에 대해서는 이미 앞에서 언급하였으므로 여기서는 구체적인 설명을 하지 않는다. 새로운 주구역에 필요한 값들을 설정하여 주고 다시 한번 값들을 확인하여 본 다음 《창조》단추를 누르면 새로운 주구역이 창조된다.

앞의 실례에서와 같이 《sec.》라는 령역을 관리하는 주구역을 창조하였다면 시작대면부에 《sec》령역의 항목이 나타난다.

이와 같은 방법으로 역방향구역 다시 말하여 IP로부터 령역이름을 찾도록 관리하여주는 구역을 만들수 있다.

이렇게 《sec》구역과 《192.168.8》구역을 창조하였으면 설정항목을 리용하여 필요한 관리를 진행할수 있다.

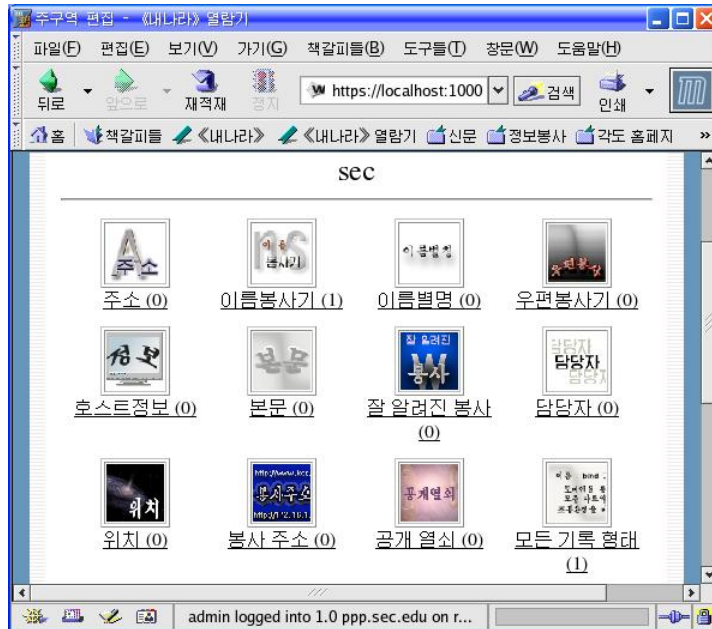



그림 3-13. 《sec》구역의 관리대면부

령역이름봉사기관리도구항목에 나온 《sec》구역을 찰칵하면 그림 3-13과 같은 관리도구창이 현시된다. 이 창문을 리용하여 필요한 정보를 추가 또는 삭제하여주는 방법으로 이름령역을 관리할수 있다.

여기에 나오는 설정항목들에 대해서는 이미 앞에서 설명한 파일에 의한 설정에서 언급된 내용들이므로 더 설명하지는 않는다. 그러므로 앞에서 설명한 령역이름봉사기관리부분에 대해서 잘 파악하여야 관리도구를 리용한 설정과 관리를 잘 할수있다.

여기서 이 령역에 필요한 정보를 추가하려고 하면 그림 3-14와 같은 대화창이 나타난다. 그림에서도 알수 있는바와 같이 sec구역에 주소를 추가할수 있다. 이러한 과정을 거쳐 령역을 구축하고 이름봉사기의 역할을 할수 있게 준비된다.

우와 같은 방법으로 역방향주소령역의 설정과 관리도 진행할수 있다.

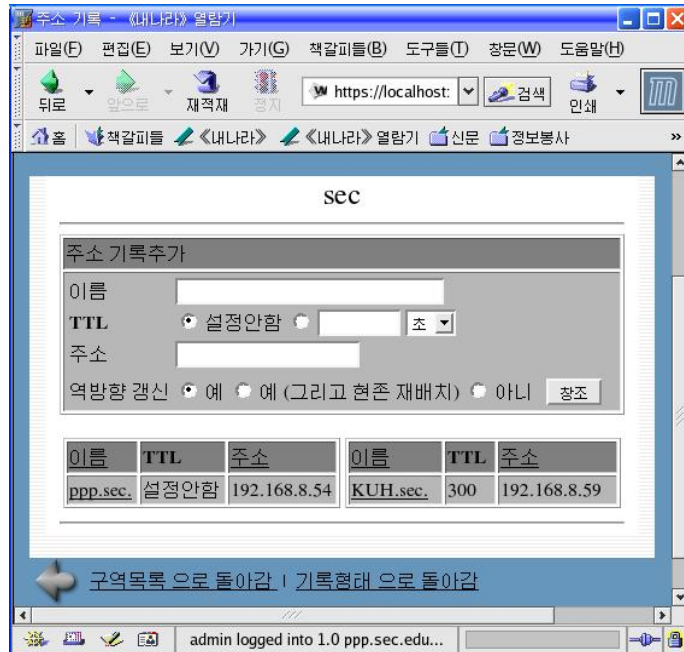



그림 3-14. sec구역의 기록추가실례

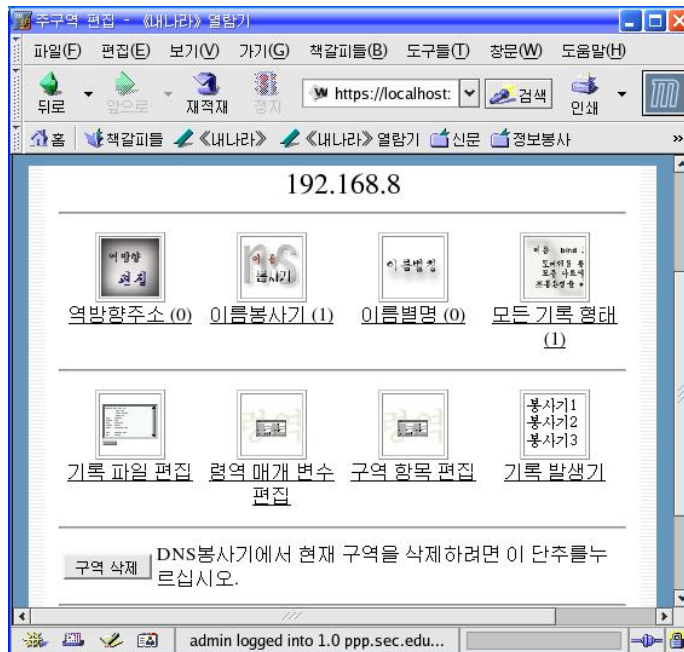



그림 3-15. 192.168.8관리대면부

그림 3-16. 192.168.8령역의 주소추가대면부

《192.168.8》구역의 편집을 설정하면 그림 3-15,16과 같은 대면부창문이 나타난다. 그의 사용과정은 앞에서 설명한 sec구역의 설정과 관리와 같으므로 설명은 생략한다. 이러한 과정을 거치면 령역이름봉사기의 설정이 기본적으로 되였다고 볼수 있다.

령역이름봉사기의 설정이 기본적으로 되였다면 그림 3-17과 같은 대면부에서 《이름봉사기 기동》을 찰칵하면 된다. 오유통보문이 나타나지 않으면 령역이름봉사기가 안전하게 기동되였다는것을 의미한다.

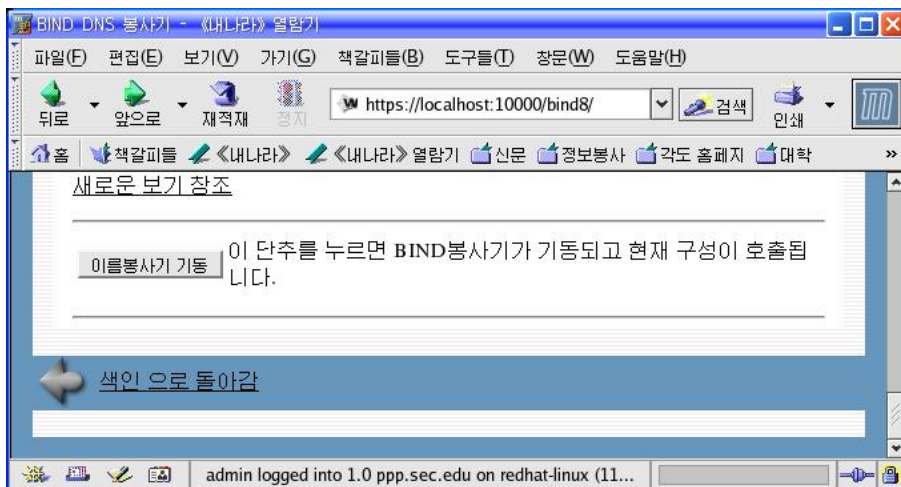



그림 3-17. 이름봉사기의 기동

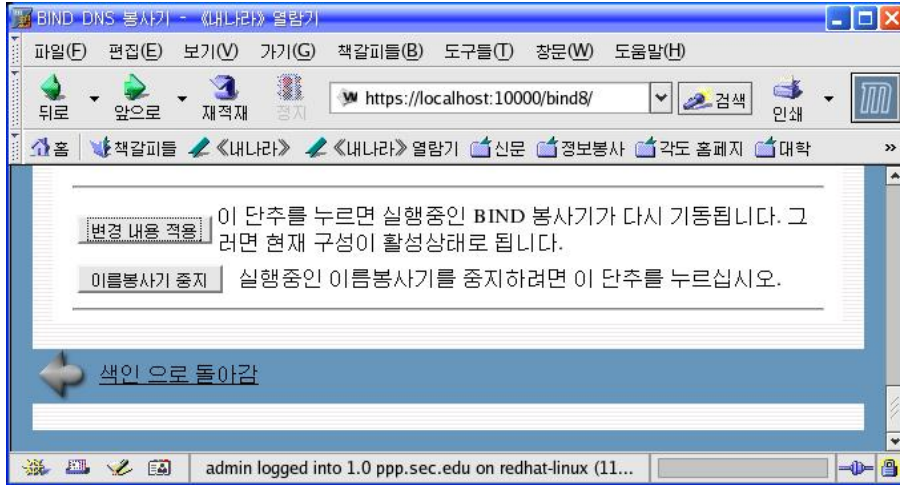



그림 3-18. 이름봉사기의 중지와 재기동

령역이름봉사기의 설정내용 다시 말하여 령역내용이 추가되였거나 삭제 혹은 달라졌다면 변경내용을 보관하고 봉사기를 재기동시켜야 한다.

그렇게 해야 변경내용이 적용되여 봉사기가 변경된 내용대로 봉사를 진행할수 있다.

봉사기에서 변경내용의 적용과 중지, 재기동은 그림 3-18과 같은 과정을 거쳐 실행시킬수 있다.

이상과 같은 과정을 거치면 령역이름봉사기(일명 DNS봉사기)를 설정하고 관리할수 있으며 봉사기의 기동과 정지를 할수 있다.

여기서도 중요하게 언급할 문제는 봉사기관리도구를 리용하여 봉사기의 관리를 진행하면 그 내용들이 모두 앞에서 설명한 파일들에 보관되며 그 파일들에 의하여 봉사기가 봉사를 진행한다는것이다.

때문에 봉사기의 관리를 진행하는데서 파일들의 역할과 그의 동작과정을 잘 아는것이 매우 중요하다.

제4장. DHCP봉사기

이 장에서는 DHCP에 대한 개념과 DHCP관리자로서 문제가 제기되였을 때 어떻게 처리하여야 하는가에 대하여 기본중심을 두고 설명을 진행하게 된다.

그러면 DHCP라는것이 무엇인가?

DHCP(Dynamic Host Configuration Protocol)라는것은 망파라메터의 자동설정을 진행하는 의뢰기/봉사기형의 통신규약이다. 매개 의뢰기에 기동시 동적으로 IP주소를 할당하고 기정경로와 같은 망파라메터의 통지를 진행하며 끌낼 때에는 할당한 IP주소를 회수한다. 봉사기측에서는 IP주소를 DHCP의뢰기용으로 몇개 종합하여 준비하여 놓기만 하면 된다. 동시에 관문주소나 령역이름, 부분망가리개 등의 정보를 의뢰기에 통지할수도 있다. 이에 의하여 콤퓨터가 부분망에서 이동하여도 IP주소나 망가리개 등 파라메터들의 설정에 대하여 생각하지 않아도 통신을 할수 있다.

이것은 콤퓨터망의 리용과 관리에 대단히 유리한 조건을 지어준다. 이에 대한 구체적인 설명은 아래에 내려가면서 진행하기로 하자.

이번 장에서는 다음과 같은 체계로 설명을 진행하게 된다.

- DHCP봉사기에 대한 간단한 리해
- DHCP봉사기의 설치와 구축
- 관리도구를 리용한 DHCP봉사기의 구축과 관리

제1절. DHCP봉사기에 대한 간단한 리해

DHCP봉사기란 간단히 말하여 말단기(의뢰기)들에 IP주소를 할당해주는 봉사기를 말한다. Linux나 Windows조작체계를 사용하여 콤퓨터망에 접속하는 콤퓨터들은 모두 TCP/IP규약을 사용한다는데 대해서는 이미 앞에서 설명하였다. TCP/IP규약을 사용하는 모든 콤퓨터들은 자기의 고유한 IP를 가진다는데 대해서도 앞에서 언급되였다. 그러므로 이 세상의 모든 콤퓨터들이 콤퓨터망에 들어가자면 다른 콤퓨터와는 구별되는 다시말하여 자기만의 고유한 IP주소를 가져야 한다. 그런데 정보산업의 시대가 도래하면서 콤퓨터의 대수가 급격히 늘어나고 콤퓨터망의 리용률이 높아지면서 사람들은 IP의 부족을 느끼기 시작했다. 다시말하여 이전에는 생각하지도 않았던 문제 즉 IP자원의 부족이라는 도전에 부닥치게 된것이다. 처음 TCP/IP규약을 만들 때 사람들은 이만한 IP이면 전 지구상의 인구가 쓰고도 남으리라고 타산하였다. 그러나 콤퓨터에 의한 정보산업의 시대는 IP부족이라는 심각한 문제를 산생시켰다. 또한 콤퓨터의 리용이 일반화되고 많은 사람들이 콤퓨터를 리용하여 업무를 처리하면서 이전에는 반드시 설정하여야 하는것으로 되여있는 IP주소의 할당이 문제로 제기되였다. 한마디로 IP주소를 할당하는데서 반드시 자기만의 IP주소를 할당하여야 하는데 전문가도 잘 모르는때가 많은 이러한 문제는 콤퓨터의 광범한 리용에서 문제로 제기되였던것이다. 또한 콤퓨터를 다른곳으로 가지고 이동하면서 작업하는 일이 생길때마다 매번 IP를 바꾸어야 하는데 이것은 매우 시끄러운 일로 제기된다.

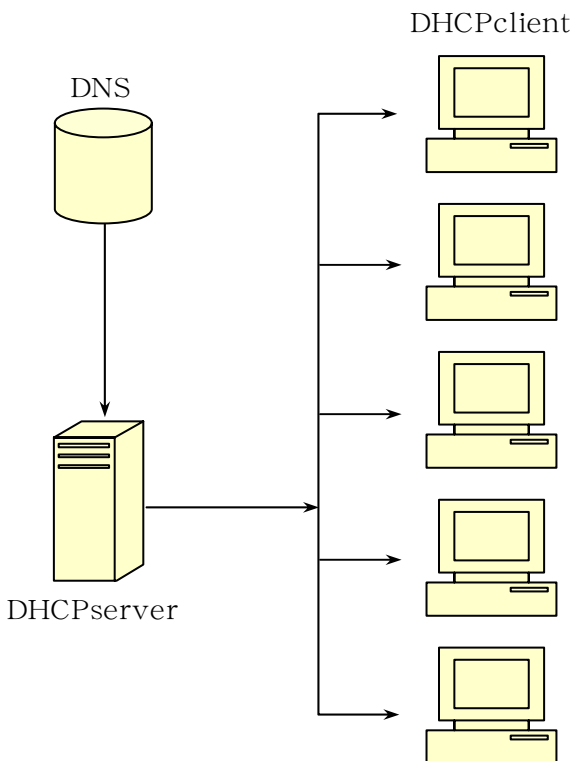
이러한 여러가지 문제들을 해결하기 위하여 세상에 나온것이 DHCP봉사기이다.

그림 4-1에서 알수있는바와 같이 DHCP봉사기는 의뢰기들로부터 의뢰를 받으면 자기에게 할당되여 있는 IP주소들가운데서 리용하지 않는 IP주소를 그 의뢰기에 할당하여 주고 의뢰기의 작업이 끝나면 IP를 다시 해방시켜준다.

그림 4-1. DHCP 봉사기의 동작구성

이러한 과정을 수행하는 DHCP봉사기는 다음과 같은 우점을 가진다.

- TCP/IP구성의 간소화

DHCP봉사기를 사용하면 TCP/IP를 설정하는 시간과 로력을 크게 줄일수 있다. 사용자는 자기의 IP주

소를 알 필요가 없으며 콤퓨터를 켜기만하면 된다.

- 뛰여난 확장성

DHCP봉사기를 사용하면 IP주소공간을 더 효률적으로 사용할수 있다. IP주소를 수동으로 설정하면 하나의 IP에 한대의 콤퓨터밖에 할당할수 없다. 그러나 DHCP를 사용하여 의뢰기의 IP주소와 할당하는 시간을 정하면 콤퓨터를 사용하지 않을때에는 다른 콤퓨터가 이 IP를 사용할수 있다. 한마디로 적은 수의 IP를 가지고도 많은 의뢰기들이 사용할수 있게 한다.

- 안전한 콤퓨터망의 운영

넓은 IP주소대역을 사용하는 콤퓨터망에서 주콤퓨터의 IP주소를 모두 수동으로 설정하면 다른 사람이 IP주소를 바꿀때 기존의 IP주소와 충돌이 일어날수 있다. 그러나 DHCP를 사용하면 주소충돌문제를 막을수 있다.

이러한 우점들로 하여 DHCP봉사기는 대부분의 초고속인터네트봉사에서 널리 사용되고 있다.

그러면 DHCP봉사기의 동작과정에 대해 보도록 하자.

① 의뢰기가 기동하면 자동적으로 DHCP검색파케트를 국부망전체에로 방송(broadcast)하여 사용자의 존재를 알려주고 DHCP봉사기를 찾는다. 의뢰기가 방송을 진행하는 리유는 아직 자기의 IP주소도 모르고 DHCP봉사기의 주소도 모르기때문이다. 방송하는 파케트의 원천주소는 《0.0.0.0》이며 목적주소는 《255.255.255.255》이다. 또한 의뢰기는 사용자의 MAC주소를 포함해서 전송을 진행한다.

② DHCP봉사기는 의뢰기에서 발생된 DHCP탐색파케트를 수신하면 이에 대한 응답으로 DHCP OFFER파케트를 전송한다. 이 파케트에는 의뢰기의 MAC주소, DHCP봉사기가 관리하는 IP주소중에서 임대하여 줄수있는 IP주소, IP주소의 부분망마스크, 임대기간 등이 포함되여있다. 이 통보문도 방송을 리용하여 전송된다. 그 리유는 아직 의뢰기에 IP주소가 할당되지 않았기때문이다. 의뢰기는 파케트에 있는 MAC주소를 리용하여 주소가 같으면 이 파케트를 수신한다. 의뢰기는 DHCP봉사기들에서 받는 파케트중에서 가장 먼저 받은것을 선택한다.

③ 의뢰기는 DHCP봉사기에서 받은 OFFER통보문중에서 제일 먼저 도착한것을 선택하고 이 IP주소를 사용하기로 결정한다. 그리고 이 내용을 다시 방송으로 전체 콤퓨터망에 알려준다. 이 파케트에는 DHCP봉사기의 IP주소와 자기가 사용할 IP주소정보가 들어있다. 파케트를 방송하는 리유는 만약 DHCP봉사기가 여러대 있을 때 이들에게 알려주어야 IP주소의 중복사용을 막을수 있기때문이다.

④ 이 파케트를 수신한 DHCP봉사기는 DHCP ACK통보문을 사용해서 의뢰기가 이 주소를 사용하는것을 허락한다. 이때도 아직 의뢰기의 IP주소가 결정이 안되였기때문에 DHCP ACK파케트를 방송한다. 이 파케트를 받은 다음 의뢰기의 TCP/IP정보가 초기화되고 TCP/IP통신을 할수 있다.

이와 같은 과정을 거쳐 의뢰기는 IP를 할당받으며 의뢰기는 이 IP를 리용하여 통신을 진행한다. 그러면 DHCP봉사기의 설치와 구축, 관리에 대해 보기로 하자.

제2절. DHCP봉사기의 설치와 구축

4.2.1. DHCP봉사기의 설치

설치과정은 이미 앞에서 설명한 봉사기들에서와 같이 두가지 방식 즉 RPM에 의한 설치와 원천파일에 의한 설치가 있다. 앞에서 자세히 설명한 내용들이므로 설치과정에 대해서는 간단히 RPM에 의한 설치에 대해서만 보기로 하자.

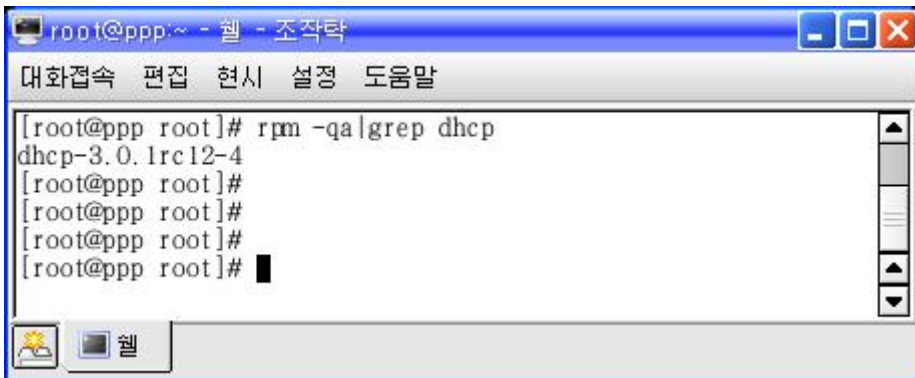



그림 4-2. DHCP의 설치확인지령

RPM에 의한 설치과정은 이미 앞에서 본것과 같이 아래와 같은 지령을 리용하여 진행한다.

```
#rpm -ivh dhcp*
```

설치과정이 무사히 진행되였다면 그림 4-2와 같이 제대로 설치되였는가에 대해 알아볼수 있다. DHCP봉사기의 설치를 진행한 다음에는 DHCP봉사기가 어떻게 구성되여 있는가에 대해 아는것이 중요하다. 왜 그런가 하는데 대해서는 앞에서 이미 언급되였기 때문에 더 설명하지 않는다.

등록부의 구성체계는 그림 4-3과 같은 방법으로 확인하여 볼수 있다.

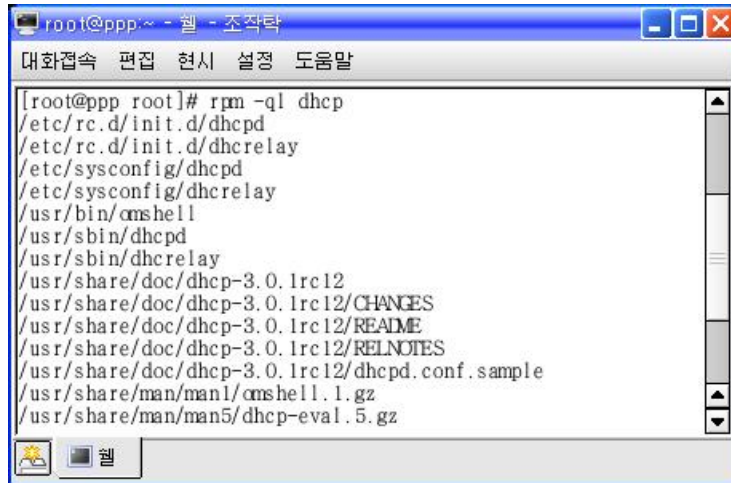



그림 4-3. DHCP의 등록부구성확인

4.2.2. DHCP봉사기의 구축

etc Rc.d Init.d dhcpd

DHCP 를 기동하기 위한 script

Dhcpd.conf

기본설정파일

usr sbin dhcpd

Dhcp 데몬

var lib dhcp Dhcpd.leases

Log 정보가 기록된 파일

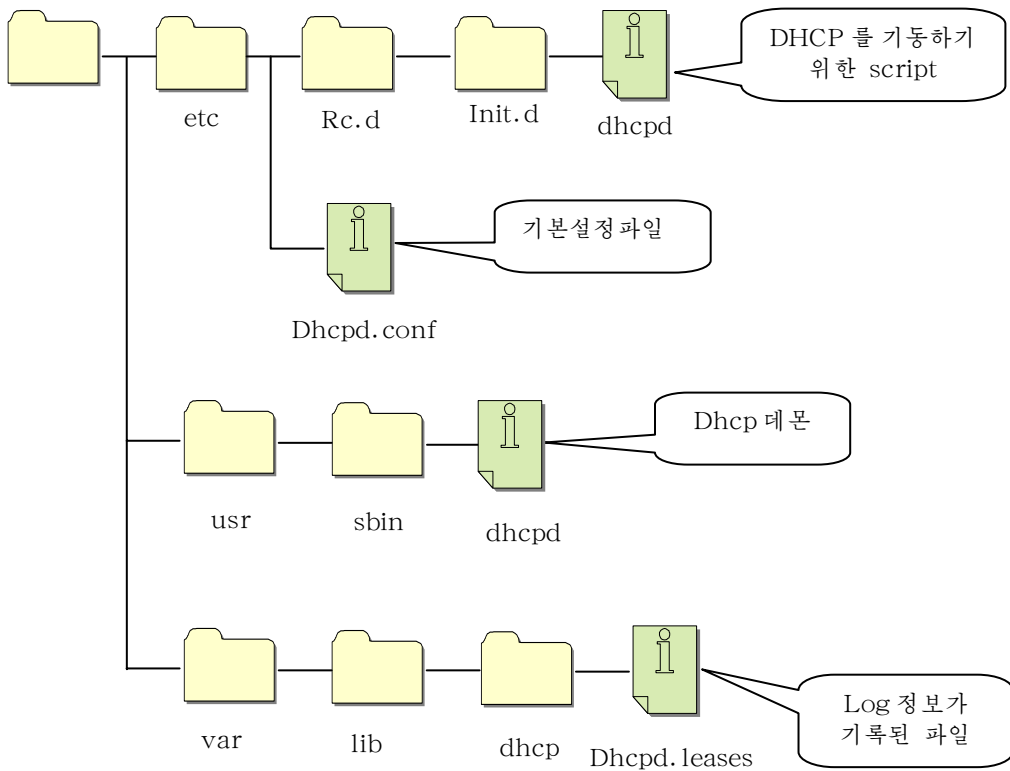



그림 4-4. DHCP 봉사기의 등록부구조

DHCP봉사기의 등록부구조와 내용에 대하여 그림 4-4에 간단히 주었다. 여기로부터 봉사기가 설치된 등록부구조와 거기에 설치된 파일들의 리용에 대하여 알수있다.

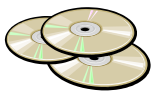
DHCP에는 의뢰기측과 봉사기측사이에 서로 주기적으로 IP를 제공하고 제공받는 관계가 형성된다. DHCP봉사기는 지정된 DNS봉사기에서 할당할수 있는 IP에 대한 콤퓨터이름(hostname)을 가지고 온다. 그다음 의뢰기측은 기동시에 IP를 받기 위해 요청을 진행한다. 그러면 봉사기에서 의뢰기의 요구를 받아들여 의뢰기가 사용할 모든 정보를 제공해주는것이다. 이러한 방식으로 가동한다는것을 알고 설정을 진행하도록 하자.

- DHCP 봉사기구축의 기초

DHCP봉사기의 설정파일은 보통 다른 봉사기의 설정파일들처럼 dhcp.conf라는 이름으로 지정되는데 /etc등록부아래에 위치하고있는것이 보통이다.

그러면 구축의 첫 과정으로 dhcpd.conf파일에 대해 보도록 하자.

보통 이 파일은 처음에는 설정되여있지 않으므로 직접 편집을 하여야 한다.

그러면 dhcpd.conf파일에 대해 보도록 하자.

실례 4-1: dhcpd.conf파일의 일부분(1)

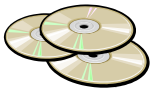
```
subnet 192.168.8.0 netmask 255.255.255.0 {
   option routers       192.168.8.1;
   #routers는 관문(gateway)를 의미하는것이다.
   option subnet-mask255.255.255.0;
   #subnet-mask는 의뢰기가 받아갈 netmask를 말한다.
   option domain-name-servers      192.168.8.54;
   #이름봉사기의 IP를 말한다.
   option time-offset   -5;
   range dynamic-bootp      192.168.8.1 192.168.8.253;
   #동적으로 할당하여 줄 IP의 범위를 말한다
   default-lease-time  600;
   #초단위의 IP할당시간이다.
   #이 시간이내에 응답이 없으면 아래의 max-lease-time으로 넘어간다.
   max-lease-time 7200;

#  option broadcast-address 192.168.8.255;
#  dns-update-style ad-hoc;
}
```

우의 실례는 한개 부분망의 기본적인 설정만을 보여주고 있다.

우의 실례에 대해 본다면 먼저 사용할 subnet의 범위를 지정해준 다음 령역이름봉사기, IP주소 할당범위, 시간 등을 각각 설정해준다. 우의 내용을 설명하면 먼저 192.168.8.0부터 netmask 255.255.255.0의 범위안에서 의뢰기에게 동적으로 IP를 할당한다는것이다.

또한 의뢰기가 사용할 령역이름과 domain-name-sever로써 읽어올 봉사기의 IP를 적어준다. 다음에 나오는 range는 동적으로 할당할 IP대역을 지적한다. 그리고 초단위로 할당할 시간값들을 적어준다.

우의 실례에서는 동적으로 할당할 IP대역을 192.168.8.1부터 192.168.8.253까지 설정하였으며 응답시간을 600초와 7200초로 설정하였다.

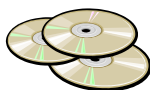
또 다른 설정실례를 보도록 하자.

실례 4-2: dhcpd.conf파일의 일부분(2)

```
option domain-name  "xxx.xx.xx" ;
option domain-name-servers 192.168.8.33;
option subnet-mask 255.255.255.0;
default-lease-time 600;
max-lease-time 7200;
subnet 192.168.1.0 netmask 255.255.255.0 {
range 192.168.1.5  192.168.1.10;
range 192.168.1.50  192.168.1.59;
option broadcast-address 192.168.1.255;
option routers 192.168.1.1;
deny unknown-clients;
}
```

우의 실례에서는 령역이름과 DNS봉사기의 IP를 《option》으로서 지정하였다. 그리고 의뢰기측에 전달할 netmask와 할당할 시간을 먼저 설정하였다. 그다음 subnet를 지정하고 range로 두개의 IP범위를 지정하였다. 이러한 설정을 진행한 다음 다시 《option》으로 broadcast와 routers로 해당 관문을 지정하였다.

마지막에 있는 deny항목은 인증되지 않은 의뢰기는 거부한다는 뜻이다.

우의 실례들을 보면서 한가지 생각되는 문제가 생긴다. 그것은 여러개의 부분망을 정의하는 경우에는 conf파일을 어떻게 설정하겠는가 하는 문제이다.

이것을 위하여 다른 실례를 하나 더 보도록 하자.

실례 4-3: dhcpd.conf파일의 일부분(3)

```
option domain-name  "xxx.xx.xx" ;
option domain-name-servers 192.168.8.33;
option subnet-mask 255.255.255.0;

subnet 192.168.8.0 netmask 255.255.255.128 {
range 192.168.8.5  192.168.8.10;
range 192.168.8.50  192.168.8.59;
option broadcast-address 192.168.1.255;
option routers 192.168.8.128;
deny unknown-clients;
}
subnet 192.168.8.128 netmask 255.255.255.128 {
range 192.168.8.150  192.168.8.180;
option broadcast-address 192.168.8.255;
option routers 192.168.8.128;
deny bootp;
}
```

우의 실례에서 보다싶이 두개의 subnet을 선언하여 사용할수 있다. 즉 기본공통적으로 사용되는 추가선택항목들을 먼저 선언하고 기 다음 subnet와 할당하는 IP주소를 선언하여 주었다. 이렇게 구성파일을 만들어 주면 여러개의 subnet구역을 만들어 관리할수 있다. 물론 이때에도 공통적으로 사용되는 추가선택항목과 차이나는 값을 가져야 한다면 subnet를 선언한 괄호({ })안에 추가선택항목들을 선언하여 줄수 있다.

우의 실례가 다른것들과 다른 점은 bootp라는 선택항목이다. 이 항목은 bootp주콤퓨터의 요청에 응답할것인가 말것인가를 지정하는것이다.

- Mac주소와의 넘기기

망기판에는 그것이 만들어지면서부터 가지는 고유한 주소가 있는데 이것이 Mac주소라고 불리우는 주소이다. 이 주소는 임의로 만들어지는것이 아니라 기판이 만들어질때 개발회사에서 할당해준 고유한 번호이다. 따라서 모든 망기판에는 자기만의 고유한 주소가 잇다고 볼수 있다.

우리가 IP를 사용하여 봉사기나 주콤퓨터의 위치를 파악하는것처럼 Mac주소를 가지고 알아낼수도 있다. Mac주소를 알아내는 방법으로는 일반적으로 arp지령을 사용한다.

아래의 표에 arp지령의 사용법에 대하여 설명하였다.

표 4-1. arp의 사용법

arp -a	캐쉬에 있는 모든 host를 보여준다.
arp -s	현재 캐쉬에 있는 Mac주소의 값을 변경한다.
arp -d	현재 캐쉬에 있는 Mac주소의 값을 지운다.
arp -v	동적인 방식으로 보여준다.
arp -i	지정한 대면부의 arp를 보여준다.
arp -n	32bit로 된 IP 즉 resolving을 하지 않은 IP를 보여준다.

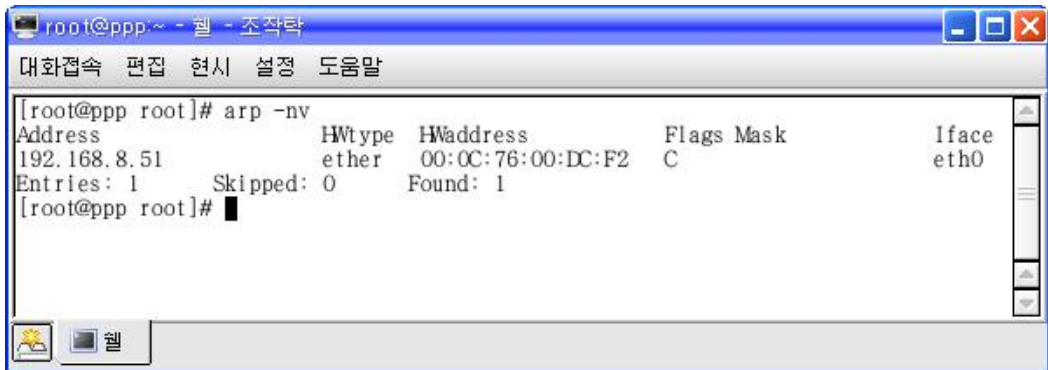
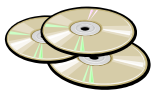



그림 4-5. arp의 사용실례

그러면 앞에서 설명한 내용에 기초하여 자기가 가지고 있는 망기판의 Mac주소를 보도록 하자. 그림 4-5가 그 실례를 보여준것이다.

우의 실례를 놓고 보면 -n추가선택항목을 주었으므로 IP주소로 주소가 나타나있다는것을 볼수 있다. 이 지령을 리용하여 해당한 망기판의 Mac주소를 알수 있으며 이것을 리용하여 DHCP봉사기의 구축시 적절한 활용을 진행할수 있다.

한마디로 특정한 주콤퓨터를 찾아가지고 그의 IP를 고정시킬수 있다.

다음의 실례를 보자.

실례 4-4: Mac주소와의 넘기기실례

```
host guest{
   hardware Ethernet 02:03:04:05:06:07;
   fixed-address 192.168.8.104;
   option host-name "guest-name.com"
}
```

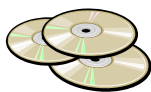
우의 실례는 Mac주소가 02:03:04:05:06:07인 망기판의 IP주소를 192.168.8.104로 고정시켜놓은 실례이다.

우와 같은 방법으로 Mac주소를 알고 특정한 기판에 IP를 대칭시켜 줄수 있다. 이러한 과정은 DHCP봉사기의 구축과 관리에서 필요한 과정이다.

이 과정을 잘 리용하면 봉사기를 보다 안전하고 안정하게 운영할수 있다.

- conf파일의 작성

그러면 앞에서 설명한데 기초하여 dhcp.conf파일을 작성하여 보자.

아래의 실례는 완전한 dhcp.conf파일이다.

실례 4-5: dhcp.conf파일

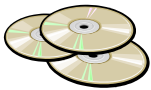
```
# test
#option domain-name "sec.edu";
#option domain-name-servers 192.168.8.54;
#option subnet-mask 255.255.255.0;
#default-lease-time 600;
#max-lease-time 7200;
ddns-update-style ad-hoc;

subnet 192.168.8.0 netmask 255.255.255.0 {
   option routers       192.168.8.1;
   option subnet-mask255.255.255.0;
   option domain-name-servers      192.168.8.54;
   option time-offset  -5;


   range dynamic-bootp      192.168.8.1 192.168.8.253;
   default-lease-time  600;
   max-lease-time 7200;

#  option broadcast-address 192.168.8.255;
#  dns-update-style ad-hoc;
}
```

이 실례에 대한 구체적인 설명은 이미 앞에서 진행하였으므로 다시 반복하지는 않겠다. 이 파일을 리용하면 DHCP봉사기를 기동시킬수 있다. 이 파일의 리용정형에 대해서는 뒤에서 보도록 하자.

4.2.3. DHCP봉사기의 기동

• dhcpd파일의 리해

봉사기로서 기동을 한다는것이 dhcpd로 기동을 한다는데 대해서는 그림 4-4를 보고 알수있었다. 그러면 dhcpd란 무엇인가 하는것이다.

dhcpd는 DHCP봉사기를 움직이기 위하여 스크립트로 만들어진 파일을 말한다.

실례 4-6: dhcpd파일

```
#!/bin/sh
#
# dhcpd          This shell script takes care of starting and stopping
#                dhcpd.
#
# chkconfig: - 65 35
# description: dhcpd provide access to Dynamic Host Control Protocol.
# Source function library.
. /etc/rc.d/init.d/functions
#먼저 공통적으로 사용되는 함수들을 불러들인다.
# Source networking configuration.
. /etc/sysconfig/network
. /etc/sysconfig/dhcpd
#
#network파일을 불러들인다.
#그것은 아래에 있는 network가 yes인지 no인지를 판별하기 위해서이다.
# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0
[ -f /usr/sbin/dhcpd ] || exit 0
[ -f /etc/dhcpd.conf ] || exit 0
[ -f /var/lib/dhcp/dhcpd.leases ] || touch /var/lib/dhcp/dhcpd.leases
RETVAL=0
prog="dhcpd"
configtest()
{
   local retval TEMP=/tmp/dhcpd$$.err
   /usr/sbin/dhcpd -t 2>$TEMP
```

```
    retval=$?
    if [ $retval -ne 0 ]
    then
            cat $TEMP
            rm -f $TEMP
    fi
    return $retval
}
start() {
    # Start daemons.
    echo -n $"Starting $prog: "
    daemon /usr/sbin/dhcpd ${DHCPDARGS}
    RETVAL=$?
    echo
    [ $RETVAL -eq 0 ] && touch /var/lock/subsys/dhcpd
    return $RETVAL
}
stop() {
    # Stop daemons.
    echo -n $"Shutting down $prog: "
    killproc dhcpd
    #해당한 프로쎄스를 찾아내여 데몬을 죽이는 역할을 한다
    #function파일에 함수로서 정의되여있다.
    RETVAL=$?
    echo
    [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/dhcpd
    return $RETVAL
}
# See how we were called.
case "$1" in
   start)
    start
    ;;
   stop)
    stop
    ;;
   restart|reload)
    configtest || exit $?
    stop
    start
    RETVAL=$?
    ;;
   condrestart)
    if [ -f /var/lock/subsys/dhcpd ]; then
```

```
          stop
          start
          RETVAL=$?
     fi
     ;;
    configtest)
     configtest
     RETVAL=$?
     ;;
    status)
     status dhcpd
     RETVAL=$?
     ;;
    *)
     echo $"Usage: $0 {start|stop|restart|condrestart|configtest|status}"
     exit 1
esac

exit $RETVAL
```

우의 dhcpd파일은 표준으로 만들어 지는 파일이므로 많은 경우 얼마 수정을 진행하지 않고도 쓸수 있다. 스크립트로 만들어져있으므로 파일에 대한 설명을 하지 않아도 리해를 하리라고 본다.

단지 dhcpd파일이 자기등록자리(보통 /etc/rc.d/init.d/)에 있는지 그리고 잘못된것은 없는지 다시 한번 확인하여 보고 DHCP봉사기를 기동시키면 된다.

- DHCP봉사기의 기동

우와 같이 모든 파일들의 설정이 제대로 되고 파일들이 자기 등록자리에 정확히 배치되였다는것을 확인하였다면 아래의 그림과 같이 봉사기를 기동시키면 된다.

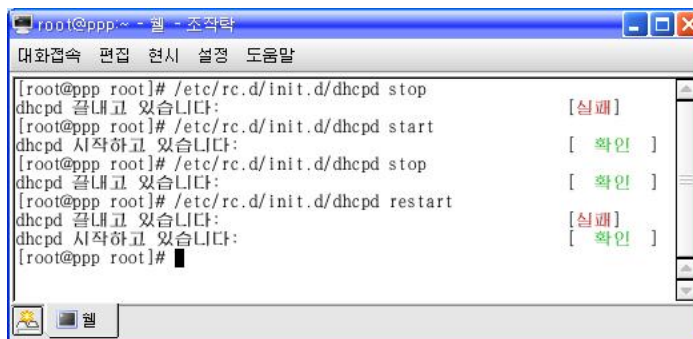



그림 4-6. DHCP봉사기의 기동과 정지

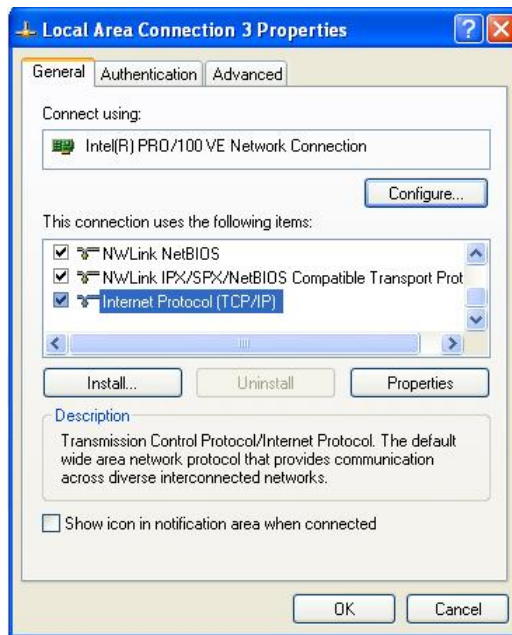



그림 4-7. InternetProtocol의 설정

여기서 InternetProtocol을 선택하고 Properties단추를 누른다.

그러면 그림 4-8과 같은 대면부가 나타난다.

여기서 자동으로 IP를 할당하는 부분을 선택을 하고 《OK》단추를 누르면 DHCP봉사기로부터 봉사를 받도록 설정이 된다.

이때 DNS봉사기나 관문의 주소는 설정하지 않아도 된다.

우와 같이 Windows의 설정과정이 끝났으면 체계를 재기동하여 DHCP봉사기로부터 봉사를 받도록 하자.

DHCP봉사기에는 dhcpd.leases라는 파일이 있는데 이 파일은 어떤 host에 어떤 IP를 할당했는가 하는 정보를 담고있는 파일이다.

아래의 그림4-9에 이 파일의 실례를 주었다.

Internet Protocol (TCP/IP) Properties

General | Alternate Configuration

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

Obtain an IP address automatically
Use the following IP address:
IP address:
Subnet mask:
Default gateway:

Obtain DNS server address automatically
Use the following DNS server addresses:
Preferred DNS server:
Alternate DNS server:

Advanced...

OK Cancel

그림 4-8. DHCP봉사기로의 설정

그림 4-9. dhcpd.leases파일의 실례

우의 실례를 보면 Mac주소가 00:0c:76:00:dd:5b인 콤퓨터에 192.168.8.57이라는 IP가 할당되였다는것을 알수 있다.

Windows의뢰기에서는 IP가 할당된것을 보려면 Command Prompt창에서 ipconfig

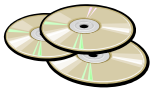
지령을 실행하면 된다.

- DNS봉사기와의 관계

우리는 앞에서 실례 4-1을 볼 때 거기에 《option domain-name-servers 192.168.8.54;》이라는 추가선택항목이 있는것을 보았다.

이것은 령역이름봉사기로는 192.168.8.54라는 IP를 가진 콤퓨터를 봉사기로 사용한다는것을 말한다.

그러면 DNS상에서는 어떻게 설정을 해주어야 하겠는가?

DHCP봉사기에서는 지정된 DNS봉사기에게 resolving을 부탁한다. 즉 요청한 주콤퓨터에 IP를 할당하기 위하여 먼저 어떤 IP를 지정할것인지 결정한 다음 DNS에 결정한 IP를 넘겨준다. 그러면 DNS는 해당한 IP를 찾아 주콤퓨터이름을 돌려준다. 이러한 방식으로 동작한다고 설명할수 있는데 그러면 DNS에 있어야 할것은 A레코드로 구성된 추가선택이 필요한것이다. 즉 IP와 대조를 시켜주어야 한다.보통 아래와 같이 주콤퓨터의 IP를 지정해준다.

실례 4-7: DNS의 zone파일의 일부분

```
xxx-184        IN   A      192.168.1.184
xxx-185        IN   A      192.168.1.185
xxx-186        IN   A      192.168.1.186
xxx-187        IN   A      192.168.1.187
xxx-188        IN   A      192.168.1.188
xxx-189        IN   A      192.168.1.189
xxx-190        IN   A      192.168.1.190
xxx-191        IN   A      192.168.1.191
xxx-192        IN   A      192.168.1.192
xxx-193        IN   A      192.168.1.193
```

DNS봉사기의 zone파일에 이러한 설정이 되여있어야 해당한 DHCP봉사기로부터 요청이 들어오면 거기에 따라 DNS봉사기와 DHCP봉사기와의 련동이 진행될수 있다.

이것을 잘 알고 해당한 봉사기를 돌릴 때 거기에 맞게 파일들의 설정을 진행하여야 한다.

제3절. 관리도구를 리용한 DHCP봉사기의 구축과 관리

앞에서도 본바와 같이 DHCP봉사기도 관리도구를 리용한 구축과 관리를 진행할수 있다. 앞에서 이미 본바와 같이 봉사기관리도구의 대면부를 펼치면 거기에 DHCP봉사기의 설치와 관리를 위한 선택항목이 있다.

이 선택항목을 선택하면 아래와 같은 창문이 펼쳐지면서 DHCP봉사기의 설정을 진행할수 있게 된다.

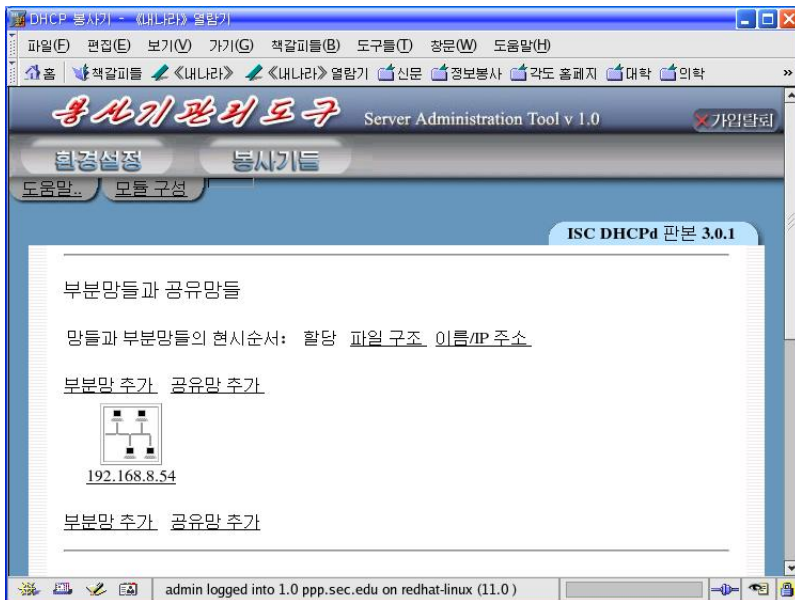



그림 4-8. DHCP봉사기관리도구의 대면부

여기서 보면 해당한 부분망들과 공유망들의 설정과 관리 및 상세한 내용의 설정들을 진행할수 있다.

아래의 그림들이 세부항목들을 보여준것이다. 그에 대한 설명은 이미 앞에서 진행하였으므로 구체적인 설명은 따로 진행하지 않는다.

그림 4-9. 부분망(subnet)의 설정

그림 4-10. 그룹세부항목의 설정

그림 4-11. 주콤퓨터세부항목의 설정

우와 같은 항목들과 세부항목들에 해당한 값을 넣어주고 필요한 설정을 진행한 다음 DHCP봉사기를 기동하면 된다.

그림 4-12. DHCP봉사기의 기동

봉사기의 기동과 정지도 관리도구를 리용하여 진행할수 있다.

관리도구의 설정이 끝나면 아래의 그림에서처럼 《봉사기시작》이라는 단추를 누르면 된다. 봉사기의 설정이 제대로 되였다면 그림 4-13과 같은 대면부가 나타나며 설정값이 틀리거나 봉사기가 돌수 없는 상태이면 해당한 오유정보를 표시하는 대면부가 나타난다.

봉사기를 운영하는 과정에 필요에 따라 설정항목을 다시 편집하고 봉사기를 운영하여야 한다면 그림 4-13과 같은 대면부를 리용하여 설정항목을 수정하고 봉사기를 다시 가동시키면 된다.

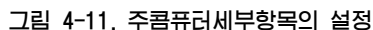
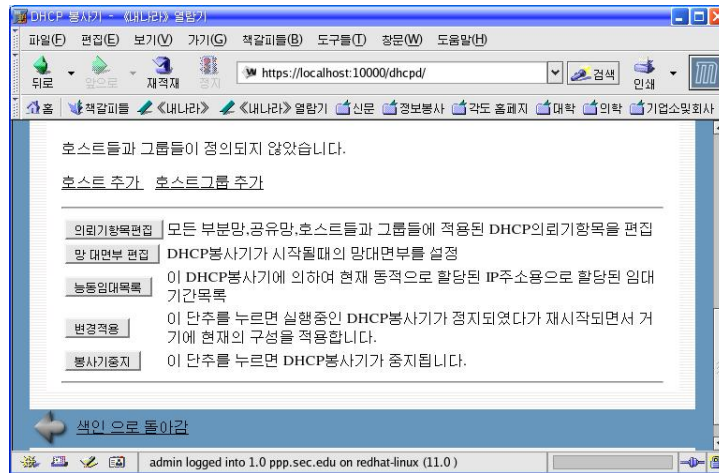



그림 4-13. 봉사기의 편집과 정지

제5장. 전자우편봉사기

《내 나라》라든가 《실리》홈페지에 들어가 본 사용자들은 누구나 《전자우편》이라는 봉사를 받아보았든가 하다못해 말이라도 들어보았을것이다.

인터네트가 사람들속에서 널리 보급되면서 많이 리용하는 봉사들가운데는 전자우편봉사도 속한다. 현재 전자우편봉사는 웨브봉사와 함께 콤퓨터봉사에서 중요한 등록자리를 차지하고있다. 전자우편봉사란 간단히 말하여 전자화된 우편봉사 다시말하여 인터네트를 통하여 사람들사이에 편지를 주고 받는것을 말한다. 과거의 우편봉사는 편지를 보낸다음 일정한 시간이 지나야 그것이 보내는 사람에게 전달되며 또 받았는지에 대한 확인을 하기도 힘들었다. 그러나 전자우편봉사기를 리용하면 임의의 장소에서 인터네트에 가입된 임의의 사람 혹은 특정한 집단 혹은 전세계를 향하여 편지를 보낼수도 있고 받을수도 있다. 또한 편지와 함께 첨부파일의 형식으로 필요한 자료를 함께 받을수 있다.

전자우편봉사기만이 가지는 이러한 우점들로 하여 전자우편봉사기는 오늘날 가장 많이 리용되는 봉사기로 되였으며 그의 리용범위는 점점 더 넓어지고 있다.

이 장에서는 현재 전자우편봉사프로그람으로 널리 쓰이고있는 Sendmail봉사기의 설정방법과 이것을 리용한 여러가지 응용에 대하여 자세히 취급한다.

이 장의 간단한 구성은 아래와 같다.

- Sendmail이란 무엇인가?
- Sendmail의 설치와 구성파일의 만들기
- Sendmail.cf파일의 환경설정
- 봉사기관리도구를 리용한 전자우편봉사기의 설정과 기동

제1절. Sendmail이란 무엇인가?

인터네트라든가 국부망들에서 전자우편(E-mail)을 전송하기 위해서는 전자우편봉사를 제공하는 우편봉사기가 필요하다.

우편봉사기(Mail Server)에는 여러가지 종류가 있지만 오래전부터 사용되여 왔으며 현재 가장 많이 사용하는 우편봉사기로서는 Sendmail이다.

그러면 전자우편이 어떤 경로를 통하여 어떻게 전송되는가에 대해 보도록 하자.

우편전송과정은 아래의 표와 같이 3개의 부분(혹은 프로그람)으로 갈라볼수있다.

표 5-1. 전자우편전송프로그람의 구분

부 분	설 명
MUA (Mail User Agent)	우편의뢰과정
MTA (Mail Transfer Agent)	우편봉사과정
MDA (Mail Delivery Agent)	우편려과와 우편의뢰기로의 우편전송과정

례를 들어 ppp.sec.edu라는 우편봉사기를 사용하는 Kim이라는 사용자가 kkk.com이라는 우편봉사기를 사용하는 Li라는 사용자에게 우편을 보낸다고 가정해보자. 먼저 Kim이라는 사용자는 Netscape Messanger나 Internet Explorer와 같은 MUA(Mail User Agent)를 리용해서 우편봉사기인 ppp.sec.edu의 25번 포구로 접속을 하게 된다. 25번 포구에서는 Sendmail이 대기하고 있다가 MUA가 보내자고 하는 우편을 받게 된다. Sendmail은 받은 우편을 /var/spool/mqueue이라는 입출력완충등록부에 보관한 다음 우편을 보낼 우편봉사기인 kkk.com의 25번 포구에 접속을 시도한다. 만약 kkk.com의 25번 포구에로의 접속이 실패하면 ppp.sec.edu의 Sendmail은 전송할 우편을 입출력완충등록부에 보관하고 수시로 접속을 시도한다. 일정한 시간이 지난 다음에도 계속 접속에 실패하는 경우 ppp.sec.edu의 Sendmail은 Kim에게 우편을 보낼수 없다는 통보문과 함께 우편을 되돌려보낸다. 접속이 되였을 경우에는 ppp.sec.edu의 Sendmail은 우편을 kkk.com의 Sendmail에 전송한 다음 입출력완충등록부의 우편을 삭제하게 된다. 우편을 받은 kkk.com의 Sendmail은 우편을 MDA로 전송하게 된다. 마지막으로 MDA는 려파과정을 거친후 해당 사용자(Li)의 우편통에 우편을 배달하는것으로서 우편의 전송과정을 모두 마치게 된다.

지금까지의 실례에서 본것처럼 Sendmail은 사용자에게서 우편을 받아 전송하여야 할 MTA에게 우편을 전송하여 주고 다른 MTA에게서 전자우편을 받아서 MDA에게 전자우편을 전송해주는 중요한 역할을 한다.

제2절. Sendmail의 설치와 구성파일의 만들기

5.2.1. RPM에 의한 설치와 등록부의 구성

앞에서 본 여러 봉사기들과 마찬가지로 전자우편봉사기의 설치에서도 RPM에 의한 설치, 원천파일에 의한 설치가 있으며 GUI도구에 의한 설치가 있다. 여기서는 RPM에 의한 설치에 대해서만 고찰한다.

Sendmail관련 rpm파일들은 배포판에 포함되여있으므로 CD를 체계에 장비한후 아래의 그림 5-1과 같이 설치를 진행하면 된다.

준비하여야 할 RPM파일들은 아래의 표와 같다.

표 5-2. 우편봉사기의 설치파일

파 일	설 명
Sendmail-8.12.5-7kp.i386.rpm	우편송신봉사기
Sendmail-cf-8.12.5-7kp.i386.rpm	Sendmail의 설정파일을 만드는 도구
Sendmail-doc-8.12.5-7kp.i386.rpm	Sendmail의 문서파일
Imap-2001a-15kp.i386.rpm	우편수신봉사기

배포판에 우의 파일들이 들어있는가를 확인하여 보고 설치를 시작한다. 설치과정은 앞에서 설명을 진행한 다른 봉사기들의 설치과정과 같다.

즉 해당한 RPM파일들을 설치를 하려는 등록부에 가져다 놓고 아래와 같은 지령을 주면 설치가 진행된다.

```
#rpm -Uvh sendmail-*
```

설치가 정확히 진행되였다면 아래와 같은 방법으로 그것을 확인하여 볼수 있다.

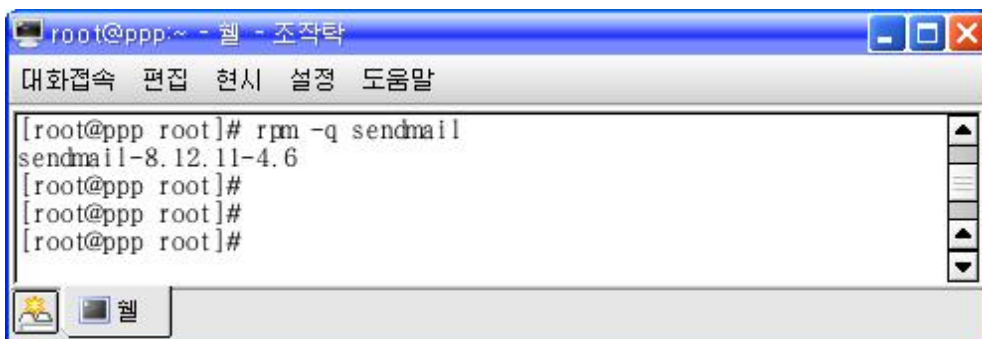



그림 5-1. Sendmail봉사기의 설치확인

설치된 봉사기의 등록부구성은 아래의 그림 5-2와 같다.

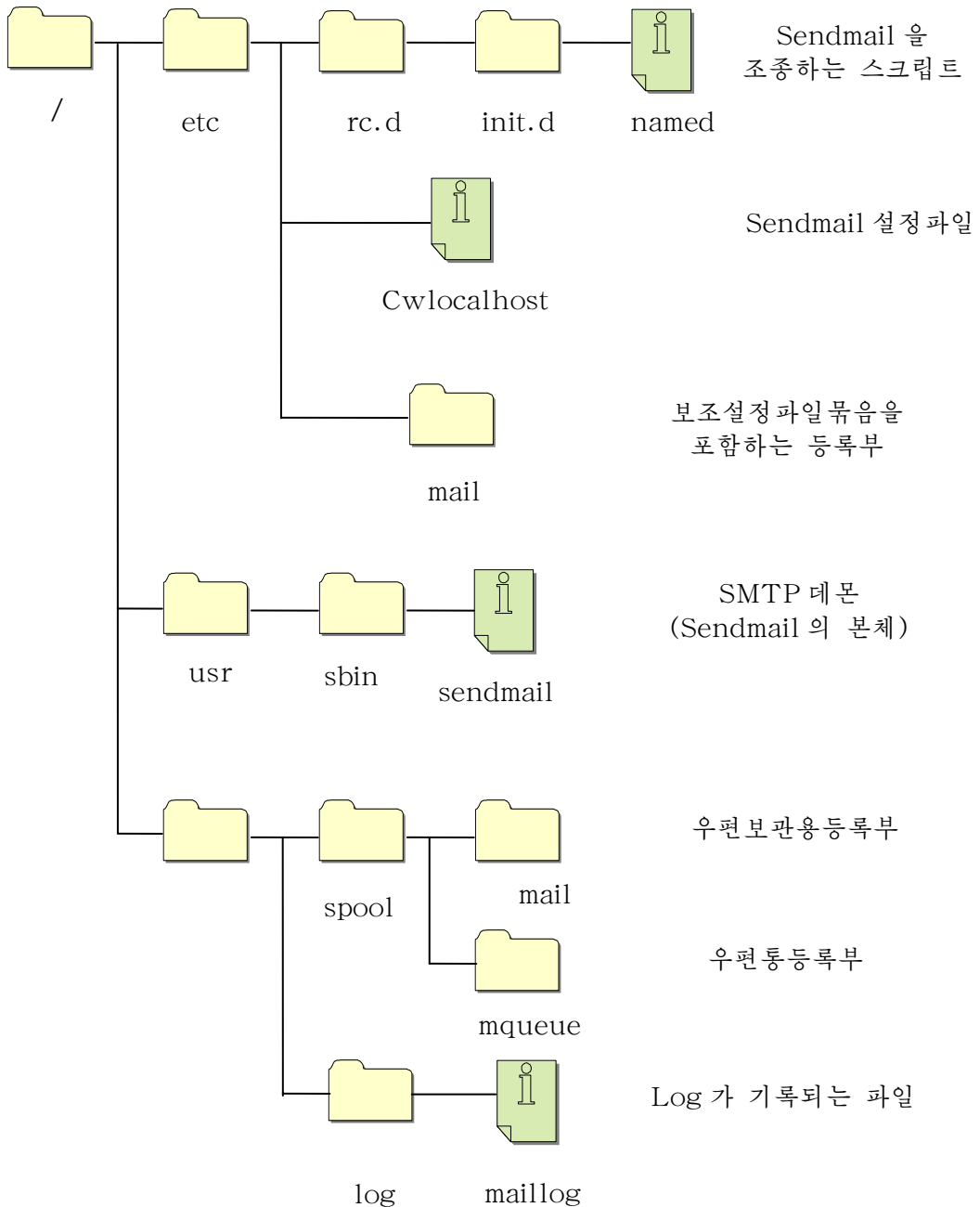



그림 5-2. Sendmail 봉사기의 등록부구조

5.2.2. 구성파일들의 만들기

그림 5-2의 등록부구성에서도 알수 있는바와 같이 Sendmail봉사기는 환경설정파일인 sendmail.cf와 access, access.db 등의 파일들로 이루어져있다. 그러면 sendmail의 동작에 중요한 영향을 미치는 몇개의 파일들에 대하여 보도록 하자.

- sendmail.cf파일

sendmail.cf파일은 sendmail의 환경설정에서 가장 중요한 파일이다.

Sendmail봉사기(이하 간단히 Sendmail)는 우편을 보내고 받을 때마다 sendmail.cf파일을 해석하여 실행을 진행한다. Sendmail이 이미 설치되여있는 우편봉사기의 sendmail.cf파일을 보면 sendmail.cf파일이 얼마나 복잡하게 구성되여있는지 알수 있다. 이전에는 이 파일을 하나하나 수동적으로 작성하였다. 그러나 지금은 Sendmail과 함께 제공되는 파일들과 m4라는 마크로처리기를 리용하여 쉽게 만들수 있다. sendmail.cf를 만드는 방법과 그의 설정에 대해서는 뒤에서 자세히 언급하기로 한다.

- local-host-names파일

Sendmail-8.9.x판본에서는 sendmail.cw라는 이름으로 되여있다.

이 파일이 없으면 Sendmail은 sendmail.cf파일의 설정이 제대로 되여있다 하더라도 정확히 동작하지 않는다.

local-host-names파일에는 우편을 받을수 있는 주콤퓨터들을 등록한다. 만약 local-host-names파일에 등록된 주콤퓨터로 우편이 전송되면 우편은 MDA(Mail Delivery Agent)로 전달된다. 그리고 만약 하나의 Linux봉사기로 전자우편봉사와 령역이름봉사를 함께 진행한다면 local-host-names에 MX레코드를 가지는 모든 주콤퓨터들을 등록해야 한다. 또한 local-host-names파일에 등록된 주콤퓨터들이 같은 우편봉사기의 IP주소를 가리키고 있을 경우에는 local-host-names에 등록된 어떠한 주콤퓨터로 우편을 전송해도 전자우편봉사기에서 우편을 받을수 있다. 자세한 파일설정은 뒤에서 언급한다.

- access 및 access.db파일

access파일은 파일의 이름에서 알수 있듯이 각종 접근설정이 보관되여있는 파일이다. 아래의 그림 5-3이 access파일의 실례이다.

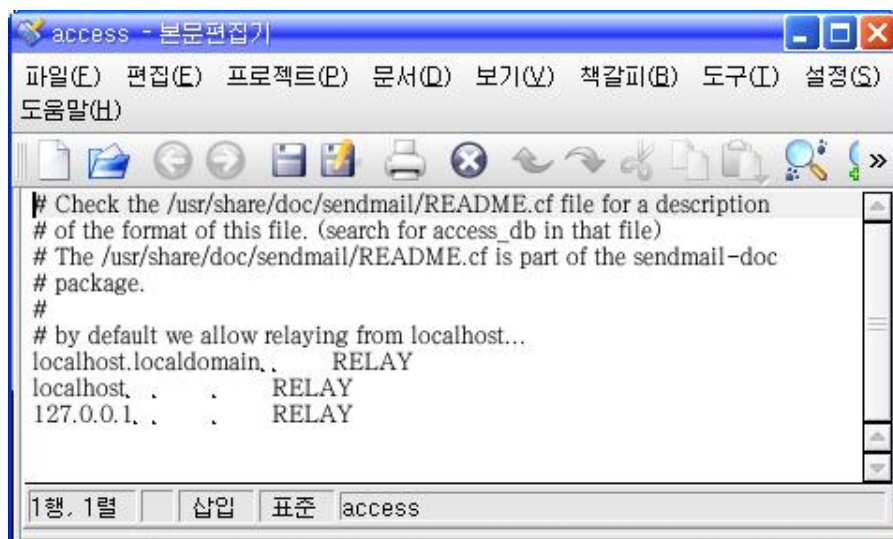



그림 5-3. access파일의 실레

실레에서 볼수 있는것처럼 이 설정은 콤퓨터에서 요청되는 모든 전자우편에 대해서 중계(RELAY)를 허용한다는것을 의미한다.

다시말하여 하나의 주콤퓨터는 다른 주콤퓨터를 SMTP봉사기로 지정하여 전자우편을 보낼수 있다. 또한 특정한 IP대역의 부분망에 있는 콤퓨터들도 주콤퓨터를 SMTP봉사기로 지정하여 우편을 보낼수 있다.

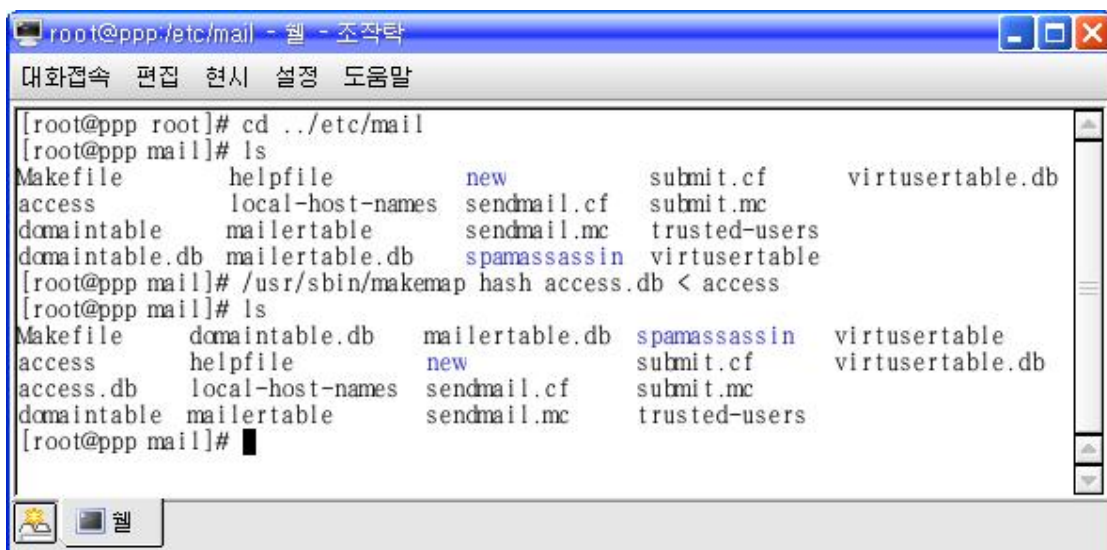



그림 5-4. access.db파일의 만들기

그림 5-3과 같이 설정할 경우 주콤퓨터와 지정된 부분망주소를 가진 콤퓨터들을 제외한 다른 곳에서는 이 전자우편봉사기를 경유하여 이 우편봉사기가 관리하지 않는 어떠한 전자우편봉사기로도 전자우편을 전달할수 없다. 또한 이 파일에 우편물을 받지말아야 할 주소라든가 받지말아야 할 전자우편을 등록하고 REJECT로 설정하여 전자우편의 수신을 거부할수도 있다.

access파일이 준비되였으면 이 파일을 가지고 access.db파일을 만들어야 한다.

그 리유는 access파일이 있다고 하여 우에서 설명한 기능을 수행하는것은 아니기 때문이다. access파일에서 필요한 설정이 다 되였으면 그것을 리용하여 봉사기가 리해할수 있는 파일인 access.db파일로 만들어주어야 한다.

그림 5-4에 access.db파일을 만드는 과정을 보여주었다.

그림5-4에서 볼수 있는것처럼 makemap를 리용해서 acceses파일에 편집한 규칙을 가지고 access.db파일을 만들었다.

참고로 말한다면 우의 지령행을 잘 기억해둘 필요성이 있다. 만약 access파일을 수정할 일이 있어 access파일을 수정하게 되면 우와 같은 지령행을 리용하여 accesses.db파일을 다시 만들어야 하기 때문이다. 그러므로 access파일을 수정하면 반드시 이 지령을 리용하여 access.db파일을 다시 만들어야 한다. 그러나 Sendmail의 데몬을 다시 실행시킬 필요는 없다.

우에서 본 파일들이 sendmail을 동작시키는데서 가장 중요한 파일들이다.

파일들을 만들고 필요한 설정이 다 끝나면 마지막으로 파일들을 관리하기 위하여 아래의 그림 5-5와 같이 권한설정을 진행하여야 한다.

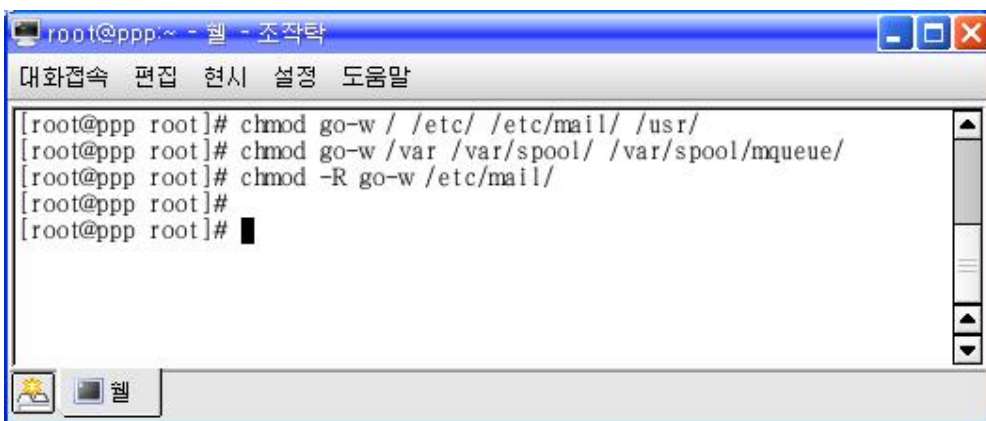



그림 5-5. 파일의 권한설정

여기에서 주의할 점은 체계의 판본에 따라 spool등록부의 위치가 다를수 있으므로 자

기의 체계에 맞게 설정을 하여야 한다.

지금까지 Sendmail의 동작에 필요한 기본적인 파일들의 만들기와 권한설정에 대하여 보았다.

마지막으로 /var/spool/등록부안에 mqueue등록부가 없다면 등록부를 반드시 만들어주어야 한다. mqueue등록부는 전자우편을 받을 봉사기 다시말하여 수신측의 봉사기가 전자우편을 받을수 없는 상태일 경우 전자우편을 림시로 보관하는 역할을 하는 등록부이므로 만일의 경우를 생각하여 이 등록부가 반드시 있어야 한다.

제3절. sendmail.cf파일의 환경설정과 POP봉사기의 설정

앞에서도 언급한것처럼 sendmail.cf파일은 전자우편봉사기에서 매우 중요한 파일이다. 그것은 sendmail.cf파일이 sendmail봉사기의 일체 동작을 규정하는 역할을 진행하는 파일이기때문이다.

sendmail.cf파일을 크게 7개의 부분으로 나누어 고찰할수 있는데 특별한 마크로를 사용하기때문에 설정이 매우 복잡하다. 이 절에서는 먼저 sendmail에서 사용되는 규칙과 7개 부분의 기능, 그리고 사용되는 마크로들에 대해서와 그의 설정에 대하여 취급한다.

5.3.1. Sendmail에서 사용되는 규칙

Sendmail은 전자우편을 받을 때마다 sendmail.cf파일을 읽어들여 여기에 지적된 규칙대로 실행을 진행한다. 그러므로 sendmail.cf파일은 빨리 읽고 해석하기 쉽도록 표 5-3과 같이 일정한 규칙을 가지고 작성되여있다.

표 5-3. sendmail의 규칙

번 호	규 칙
1	Sendmail파일은 파일의 줄단위로 실행된다.
2	모든 지령행은 config지령이거나 #문자로 시작해야 한다.
3	Sendmail파일의 줄은 공백으로 시작할수 없다.
4	Sendmail파일의 줄의 시작이 Tab로 시작되면 우의 줄의 련속이다.
5	Sendmail파일의 줄에는 반드시 하나의 지령이 존재한다.
6	Config지령은 변수로 시작된다.

우의 표에서 알수있는것처럼 sendmail.cf파일은 config지령으로 구성된다는것을 알수있다. 표 5-4는 sendmail.cf파일에서 제공하는 config지령과 그에 대한 간단한 설명이다.

표 5-4. config지령

Config지령	의　미
V	Sendmail.cf파일의 준위
M	MDA지정
D	마크로지정
R	다시 쓰기 (Rewrite)의 지정
S	Rule-set지정
C	Class지정
F	Class macro from a file or a pipe
O	Option
H	Header
P	Delivery priorities
T	Trusted usrs
K	Key database
E	환경변수의 지정
L	Include extended load average suport

표 5-5. config내부마크로

내부마크로	의미
a	RFC822형식에 의한 발신날자
b	RFC822형식에 의한 현재날자
c	HOP 계수
d	Unix ctime형식의 날자
e	SMTP접속통보문
f	‘from’을 리용한 송신자의 주소
g	수신자와 비교한 송신자의 주소
h	수신콤퓨터
i	queue id
j	공식적인 령역이름
l	Unix서식의 행
n	데몬이름
o	관리자주소

p	Sendmail의 pid
q	송신자주소의 기본서식
r	사용된 규약
s	송신자의 콤퓨터이름
t	현재시간의 수자표기법
u	수신자
v	Sendmail의 판본
w	현재 싸이트의 콤퓨터이름
x	송신자의 전체이름
z	수신자의 홈등록부

표 5-5는 config지령과 함께 쓰이는 내부마크로이다. 내부마크로는 사용자가 직접 정의하여 사용할수도 있다.

5.3.2. sendmail.cf의 구조

sendmail.cf파일은 7개의 부분으로 나누어진 매우 복잡한 구조를 가지고있다.

그러면 매개 부분들이 어떤 기능을 수행하는지에 대해 보도록 하자.

- Local Info부분

Local Info부분은 해당 주콤퓨터의 구성정보를 정의하는 부분이다. 이 부분에는 콤퓨터의 이름, 전자우편령역이름 등의 정보가 정의되며 SMTP에 접속하는 동안 Sendmail에 표시하는 통보문이나 설정파일의 판본정보 등도 정의한다.

- Option부분

Option부분은 Sendmail의 환경을 정의하는 추가선택을 정의하는 부분이다. Option부분에서 사용하는 추가선택은 100여개에 이른다. 하지만 Option부분의 추가선택항목들은 거의 모두가 체계에 최적으로 설정되여있으므로 이 부분은 수정할 필요가 없다.

- Message Precedences부분

Message Precedences부분은 통보문의 우선순위를 할당할 때 사용된다. 가장 최우선으로 할당되는 통보문은 전자우편이다. 이 부분의 설정에서 우선순위는 나타내는 수자가 높은 수자일수록 높게 된다. 하지만 우선순위가 높은것은 아무런 의미가 없다. 우선순위가 높다는것은 탐색순서를 나타내는 번호를 선택하는 일뿐이기때문이다. 레를 들어 우선순위를 나타내는 번호가 낮은 전자우편 특히 우선 순위가 부수일 경우 전자우편으로 보낼 오유통보문은 만들어지지 않는다. 그러므로 많은 량의 전자우편을 처리할 경우에는 우선

순위가 낮을수록 유리하다. Message Precedence부분도 Option부분과 마찬가지로 체계에 최적으로 설정되여 있으므로 수정할 필요는 없다.

- Trusted Users부분

Trusted Users부분은 발신인 주소를 변경할 때 사용된다. 먼저 발신인 주소를 변경할 때 Trusted Users부분에 등록되는 사용자는 체계에 등록된 사용자이여야 한다.

다시말하여 /etc/passwd파일에 존재하는 사용자이여야 한다.

Trusted Users부분에 등록된 root,daemon,uucp는 절대로 변경하면 안된다.

또한 Trsted Users부분에서 사용자를 추가하면 보안상 문제가 생길수 있으므로 일반 사용자를 추가하는것은 피하여야 한다.

- Format of Headers부분

Fomart of Headers부분은 Sendmail이 전자우편에 삽입할 머리부를 정의한다. 전자우편머리부란 전자우편의 가장 첫부분에서 볼수 있는것으로서 전자우편이 언제 어디서 왔는지 등의 전자우편관리에 필요한 정보를 말한다.

- Rewriting Rules부분

Rewriting Rules부분에는 전자우편주소를 사용자가 사용하는 전자우편프로그람이 해석하여 해당한 전자우편프로그람에서 적당한 서식으로 우편을 다시 작성할 때 필요한 규칙이 정의되여 있다.

- Mailer Definition부분

Mailer Definition부분은 Sendmail이 전자우편프로그람을 시작할 때 사용하는 지령을 정의한다. 또한 Mailer와 련관된 특수한 재작성 규칙도 정의된다.

Sendmail.cf파일은 우에서 설명한 이 7가지 부분으로 구성되여 있다. 그러면 지금까지의 설명에 기초하여 sendmail.cf파일의 설정에 대하여 보도록 하자.

5.3.3. sendmail.cf파일의 설정

sendmail.cf 파일은 앞에서 언급한것과 같이 Sendmail설정에서 가장 중요한 설정파일이다. 그러면 앞에서 설명한 7개의 부분을 중심으로 매 부분의 설정을 실례와 함께 보도록 하자.

먼저 Local Info부분을 보도록 하자.

아래의 그림 5-6은 sendmail.cf파일의 Local Info부분의 일부분이다.

Local Info부분은 마크로를 정의하는 D지령과 클라스값을 정의하는 C지령, 파일에 클라스값을 넣는 F지령, 마지막으로 정보의 자료기지를 정의하는 K지령으로 구성된다.

```
Cwlocalhost
Cwlinux.sec.com
```

우의 실례부분에서 첫행은 주콤퓨터에 관하여 전자우편을 처리하기 위한 설정이다. 다음

행은 linux.sec.com이라는 령역이름으로 전자우편봉사기를 운영하겠다는 설정이다.

전자우편봉사기의 령역이름은 여러개를 설정해도 일없다. 만약 linux-1, linux-2, linux-3이라는 3개의 콤퓨터이름으로 전자우편봉사기를 운영하려고 한다면 다음과 같이 설정하면 된다.

```
Cwlocalhost
Cwlinux-1.manazone.com
Cwlinux-2.manazone.com
Cwlinux-3.manazone.com
```

이 설정에서 사용된 W클라스는 값을 대입하라는 지령이다.

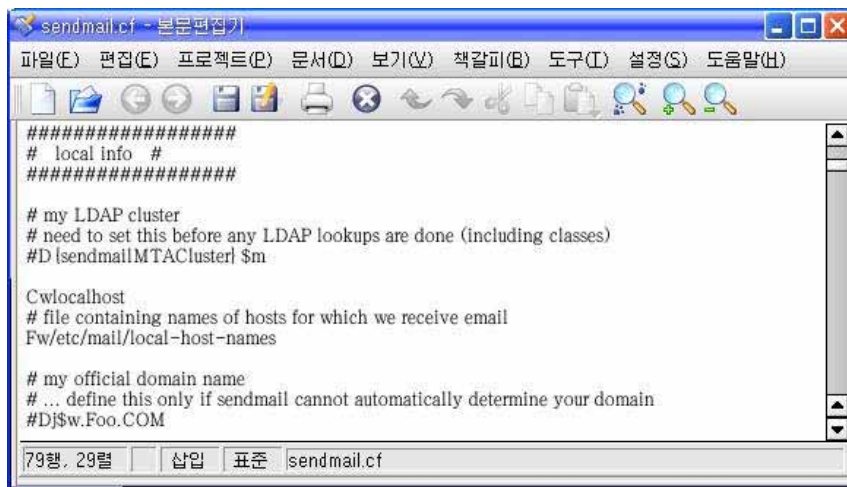



그림 5-6. Local Info부분

```
Fw/etc/mail/local-host-names
```

우의 설정은 만약 전자우편봉사기로 운영할 령역이름이 많은 경우 앞에서와 같이 매번 Cw를 사용하여 관리할것이 아니라 /etc/mail/local-host-names에 운영할 령역이름을 한줄씩 등록해서 관리하려는 경우에 사용하는 실례이다. F클라스는 지정한 파일을 읽어들인다.

일반적으로 Sendmail을 실행하는 전자우편봉사기가 다른 이름을 가진 콤퓨터의 전자우편을 받았다면 그 우편이 해당 콤퓨터에 속한것으로 인식한다. 만약 이 전자우편주소가 Fw에 지정된 파일안에 존재하면 전자우편봉사기는 그 우편을 받아들인다.

```
DMsec.com
```

우의 설정은 전자우편의 발신지를 sec.com으로 설정한것이다. 전자우편은

전자우편봉사기가 전송하지만 우와 같이 설정한다면 수신측에서 받게 될 발신지는 sec.com으로 받게 된다. 다시말하여 linux.sec.com전자우편봉사기를 리용하는 zero라는 사용자가 어떤 사람에게 전자우편을 전송했을 경우 실지 전자우편주소는 zero@linux.sec.com이지만 수신측에서는 zero@sec.com라는 주소로 전자우편을 받게된다.

```
DnMAILER-DAEMON
```

우의 설정은 Sendmail이 오유통보를 보낼때 사용자이름을 정의한것이다.

다시말하여 전자우편이 정상적으로 전송되지 않아 복귀되였을 경우 복귀된 우편을 받는 사용자는 MAILER-DAEMON이라는 사용자로부터 오유통보를 받게 된다.

```
CO @ % !
```

우의 설정은 O라는 클라스변수에 @,% 그리고 !값을 할당한것이다. 이 3개의 문자는 전자우편을 손상시키기 때문에 국부사용자이름에는 사용할수 없다.

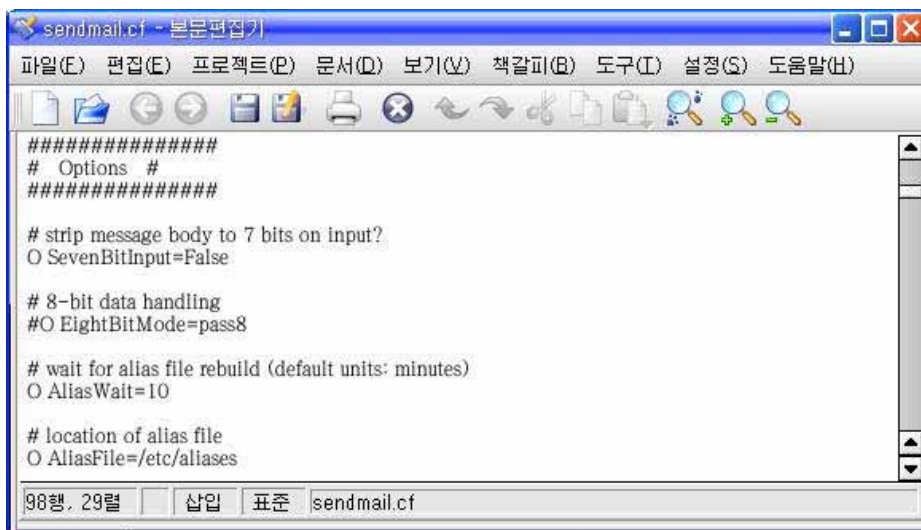



그림 5-7. Option부분

지금까지 설명한 부분이 sendmail.cf파일의 Local Info부분에서 가장 중요한 설정부분들이다. 또한 이 부분은 sendmail.cf파일에서 사용자가 정의할수 있는 유일한 부분이기도 하다.

다음으로 Option부분에 대해서 보자.

그림 5-7은 sendmail.cf파일의 Options부분의 일부이다.

Options부분은 100여개의 option으로 이루어진 상당히 긴 설정내용을 가지고있는

부분이다. 하지만 sendmail.cf파일에서 option이 어떠한 역할을 하는지 보여주는 부분의 거의 전부라고 할수있다.

중요한 역할을 하는 부분은 무엇보다도 그림에도 있는 AliasFile부분이다.

그림에서의 설정에서는 AliasFile의 위치를 etc/mail/aliases로 정의하고있다.

그러면 나머지 설정에 대해 하나씩 보도록 하자.

```
ForwardPath=$z/.forward.$w:$z/.forward
```

우의 설정은 .forward파일의 위치를 정의한 부분이다. 우의 설정에서 사용된 $z, $w값은 마크로값이다. 이 마크로들의 값은 표 5-4에서 본것과 같이 $w의 경우는 콤퓨터의 주콤퓨터이름을 나타내고 $z는 사용자의 홈등록부를 나타낸다. 또한 $w마크로에는 콤퓨터의 주콤퓨터이름이 들어있어 콤퓨터의 이름을 .forward파일의 파일확장자로 사용할수 있음을 나타내기도 한다.

실례를 들어 .forward파일은 만약 zero@linux.sec.com으로 전송되는 우편을 다른곳에서 확인하고 싶을 경우 우편을 원하는 곳으로 보낼 때 사용한다. 즉 forward파일에 zero@zeroworld.com이라고 적어주었다면 zero@linux.sec.com으로 전송되는 우편을 zero@zeroworld.com에서도 확인할수 있다.

```
Timeout.queuerturn=5d
Timeout.queuewarn=4h
```

우의 부분은 전자우편봉사기에서 전송되지 않은 우편의 대기렬처리에 대한 설정이다.

첫 행은 사용자가 요청한 우편이 Sendmail에 의해서 전송되지 못하고 5일이 지나면 undeliverable통보를 우편전송요청자에게 복귀하도록 설정한것이다. 다음 행은 Sendmail이 사용자에게서 요청된 우편을 4시간안에 전송하지 못했을 경우 우편전송요청자에게 경고통보를 전송하도록 설정한것이다.

지금까지 설명한것이 Option부분의 기능이다. 많은 경우 이 부분을 수정할 필요는 없다. 그것은 설치시에 제공되는 설정이 표준적으로 설계되여있기 때문이다.

다음으로 Message precedences부분에 대해 보도록 하자.

아래의 그림 5-8은 Message precedences부분을 보여준것이다.

이 부분도 Option부분과 마찬가지로 특별한 리유가 없다면 그대로 사용는것이 좋다.

다음으로 Trusted Users부분에 대해 보도록 하자.

그림 5-8에 sendmail.cf파일의 Trusted Users부분이 있다.

Trusted Users부분은 전자우편을 보낼때 발신인의 주소를 변경할수 있는 사용자를 설정하는 부분으로서 표 5-4에 지적된 config지령들중에서 T지령을 사용하는 부분이다.

이 부분설정에서 중요한것은 Trusted사용자가 우편을 보낼 때 송신자주소를 반복할수 있도록 인정된 사용자이므로 반드시 체계에 등록된 사용자이여야 한다는것이다. 이 부분도 역시 변경하지 않는것이 좋다.

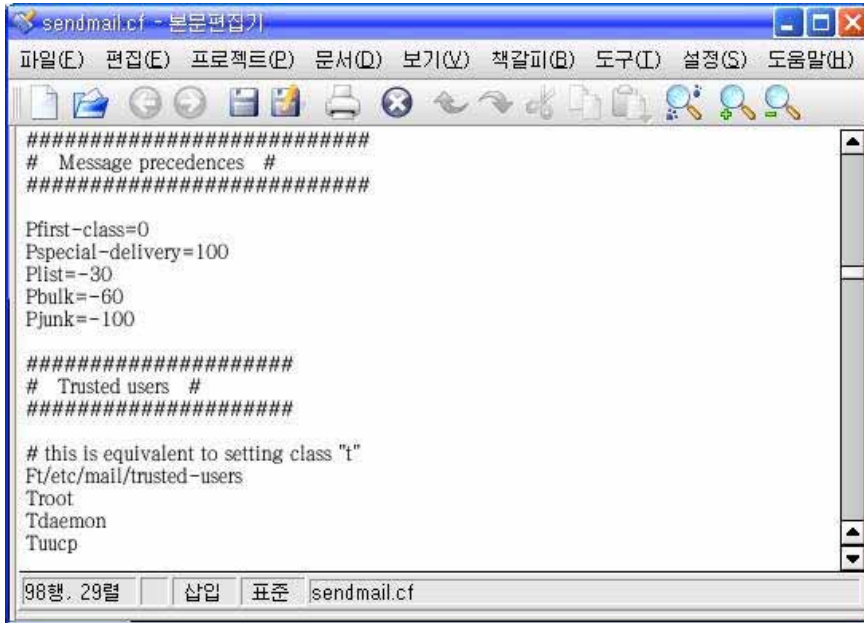



그림 5-8. Message precedences와Trusted Users부분

다음으로 Formart of headers부분에 대해 보도록 하자.

그림 5-9는 sendmail.cf파일의 Format of headers부분이다. 이 부분은 표 5-4에서 언급한 config지령들중 H지령을 사용하는 부분이다.

그림 5-9는 전송되는 우편에 삽입되는 머리부의 형식을 정의한다. 표 5-3과 표 5-4를 기초로 간단히 그의 내용을 보도록 하자.

```
H?P?Return-Path:<$g>
```

H와 P는 표 5-3에 지적된 config지령으로서 H는 머리부를, P는 Deilvery priorities를 나타낸다. 또한 Return-Path는 표 5-4에서 본 내부마크로중 수신자에 비교한 송신자의 주소를 나타내는 g를 설정하였다. 다시말하여 $g가 발신인의 전자우편주소가 들어있는 g마크로에 저장된 값을 사용하라는 설정이다.

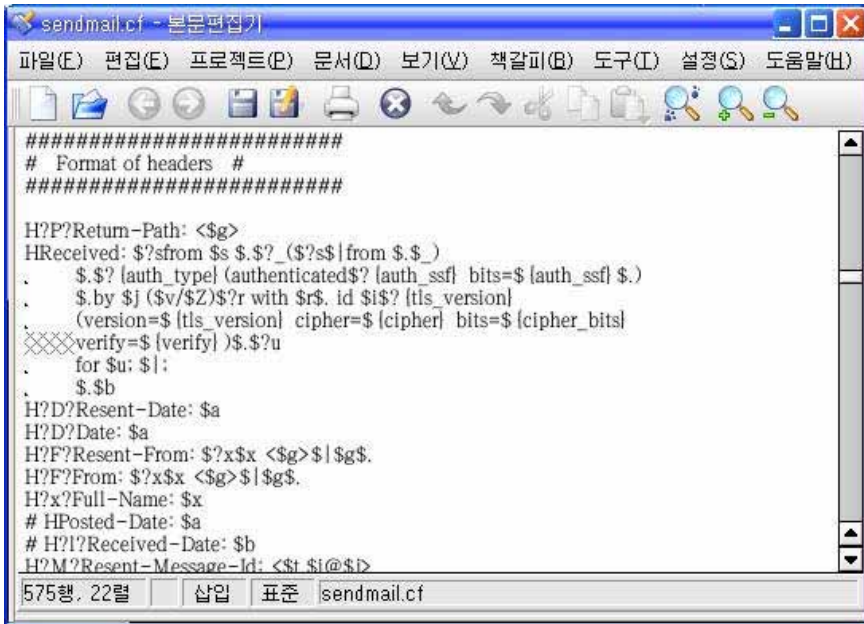



그림 5-9. Formart of headers부분

HRecieved:$sfrom $s $.$?_($?s$|from $.$)

우의 부분은 수신측에서 확인할 수신주소를 내부마크로가운데서 송신자의 주콤퓨터이름을 나타내는 s를 사용하여 설정하였다. 이 설정에서 $?는 if를 나타내고 $|는 'else'를, $.은 'elseif'를 나타낸다.

H?F?Resent-From:xx<$g>$|g.

이 부분은 마크로 x가 존재할 경우 $x<$g>를 머리부분보기로 사용하고 그렇지 않을 경우에는 $g를 머리부분보기로 사용하라는 설정이다. 따라서 매개의 경우 전송된 머리부를 비교해보면 우선 x마크로가 존재하여 $x<$g>를 리용했다면 Resent-From: Zero <zero@linux.sec.com>과 같이 나타날것이고 그렇지 않을 경우에는 x마크로가 사용되지 못하여 다음과 같은 머리부가 보일것이다.

Resent-From:zero@linux.sec.com

여기서 Zero는 송신자의 전체 이름을 의미한다.

지금까지의 설명으로 전자우편에서 우편물의 머리부가 어떻게 구성되여 전송되는가에 대하여 보았다.

다음으로 Rewriting Rules부분에 대해 보도록 하자.

그림 5-10은 sendmail.cf파일의 Rewriting Rules부분의 일부이다.

이 부분은 표 5-3에서 본 config지령들중 R지령을 사용하는 부분이다. 그림 5-10의 Rewriting Rules부분은 전자우편주소의 패턴을 다시 쓰기 위해 사용되는 규칙들을 정의한 부분이다. 다시말하여 어떤 패턴의 전자우편주소를 다른 형식으로 변환하는 여러가지 규칙들을 정의해놓은 부분이다.

Rewriting Rules부분에는 대단히 많은 재작성규칙들이 존재한다. 여기에서 한가지 패턴에 대한 주소재작성을 실례로 들어 들어보기로 하자.

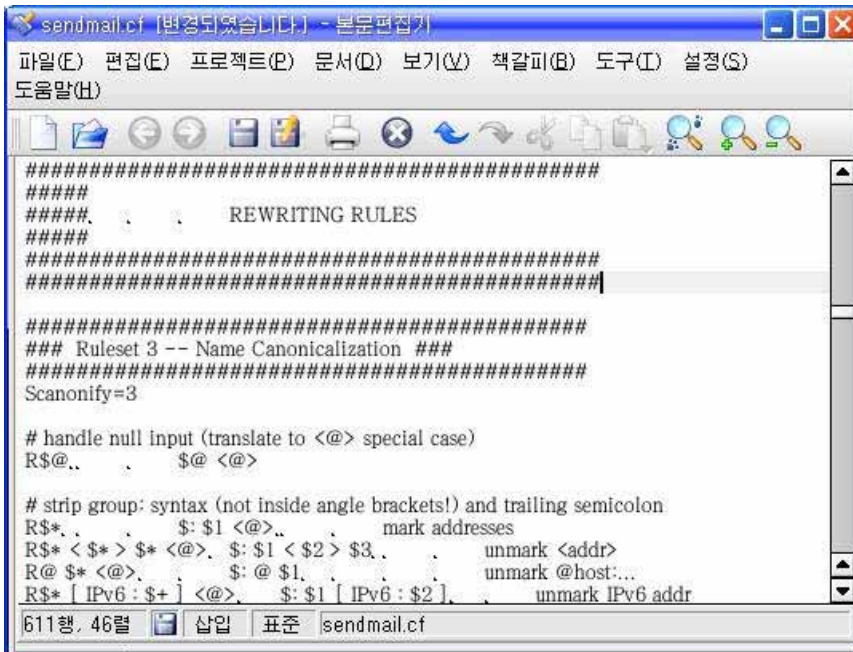



그림 5-10. Rewriting Rules부분

례를 들어 입력된 주소가 다음과 같다고 가정하자.

Zero @linux

그리고 주소 재작성 규칙은 다음과 같다고 가정하자.

R$+@$-	$1<@$2.$D>

우의 규칙에서 앞에 있는 R$+@$-부분은 재작성규칙과 비교될 입력주소의 형식을 나타내고있으며 뒤에 있는 $1<@$2.$D>부분은 재작성되여질 규칙의 형식이다. 앞에 붙은 R는 표 5-4에서 설명한 재작성지령이다.

실례에서 입력주소와 재작성되여질 규칙은 패턴과 일치한다. 먼저 재작성규칙의 입력주소규칙을 보면 '@' 문자를 사이에 두고 량쪽에 $+와 $-에 필요한 통표고리를 하나

이상 가지고있다. 또한 입력된 주소도 같은 형식이다. 또한 입력주소와 패턴주소는 모두 '@' 문자를 가지고있다. 그러므로 $+에 대응되는 문자는 zero가 될것이고 $-에 대응되는 문자는 linux가 될것이다. 그러므로 $1은 zero, $2는 linux가 된다.

이제 재작성되여질 주소의 규칙을 보자. 재작성주소의 규칙에서 이미 우리는 $1과 $2의 값을 알고있다. 나머지 $D의 값은 sendmail.cf파일에서 D마크로에 저장된 값을 대입하면 된다. D마크로에 저장된 값은 앞에서 설명한것과 같이 Local Info부분에서 확인할수 있다. 만약 D마크로에 sec.com이 저장되여있다면 재작성된 주소는 다음과 같다.

```
zero<@linux.sec.com>
```

재작성은 재작성된 주소가 다시 같은 패턴과 비교되여서야 다시 재작성될수 있다. 다시말하여 주소의 재작성이 이루어진 다음 같은 패턴이 존재하면 다시 재작성을 진행하여야 하기 때문이다. 재작성은 같은 패턴이 없을 때까지 계속 진행된다. 우와 같이 재작성된 주소는 더 이상 재작성이 이루어지지 않을것이다. 재작성 규칙은 규칙묶음으로 그룹화되여있다. 그러므로 련관성이 있는 재작성 규칙들은 이름이나 번호로 참조가 가능하다. 표 5-3에서 언급한 Rule set지정 config 지령인 S는 규칙묶음의 처음을 표시하고 이름이나 번호로 Rule set를 구별한다. 다시말하여 S4라는 지령은 Rule-set4가 시작되는 지점을 표시한다.

표 5-5는 sendmail에서 전자우편을 처리하기 위해 제공하는 규칙묶음이다.

표 5-5. 규칙묶음

규칙묶음	설 명
0	우편배달주소에 적용되여 이 주소를 우편통, 주콤퓨터, 사용자 등으로 변환한다.
1	모든 발신인주소에 적용된다.
2	모든 수신인주소에 적용된다.
3	모든 주소를 준비하기 위해 가장 먼저 호출된다.
4	내부주소형식에서 외부주소형식으로 변환한다.

주소는 발신인주소, 수신인주소, 배달주소로 되여있다. 많은 사람들이 수신인주소와 배달주소를 같다고 생각하는데 그렇지는 않다. 실례로 전자우편을 administrator@linux.sec.com으로 보내는것 같지만 실지 수신자는 root@linux.sec.com이 되는것이다. 이 두 주소는 서로 다른 규칙묶음으로 처리된다.

이러한 과정을 주소를 처리하는 규칙묶음으로 설명하면 먼저 배달주소는 규칙묶음 3으로 시작하여 우편을 배달할 전자우편프로그람과 수신인주콤퓨터의 이름, 수신인사용자이름이 포함된것으로 바뀐다. 발신인주소는 먼저 규칙묶음 3을 거쳐서 규칙묶음 1 그리고 전자우편프로그람정의에 있는 S마당을 거쳐 마지막으로 규칙묶음 4를 거쳐 재작성된

발신인주소로 바뀐다. 다음으로 수신인주소는 규칙묶음 3, 규칙 묶음 2 그리고 전자우편프로그람정의에 있는 R마당을 거쳐 마지막으로 규칙묶음 4를 거쳐 재작성된 수신인주소로 바뀐다.

다음으로 sendmail.cf파일의 Mailer Definitions부분에 대해 보도록 하자.

아래의 그림 5-11은 sendmail.cf파일의 Mailer Definitions부분의 일부이다.

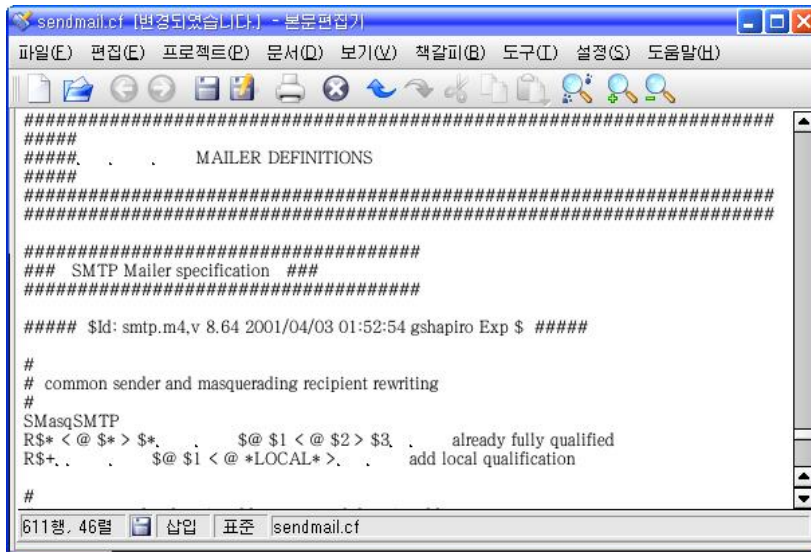



그림 5-11. sendmail.cf파일의 Mailer Definitions부분

Mailer Definitions부분은 Sendmail이 전자우편전송프로그람을 시작할 때 사용하는 지령을 정의하며 매개 전자우편프로그람과 련관된 특수한 재작성규칙도 정의한다.이 부분에서 설정해주어야 할것은 MDA(Mail Delivery Agents)를 설정해주는 다음 부분이다. 이 부분은 전자우편프로그람지령(M)으로 시작한다. 하지만 이 부분도 m4마크로로 작성되였기때문에 수정할 필요는 없다. 다만 새로운 MDA를 사용할 경우에 수정해주어야 한다.

아래에 준 실례부분에서 F=는 Flags의 략자로서 이 설정이 정의되여있으면 MDA의 동작범위가 설정된다. S=는 Sender의 략자로 앞에서 설명한 주소재작성부분에서 발신인의 주소를 재작성할 때 사용된 Rule-set부분이 정의된 부분이다.

R=는 Recipent의 략자로 역시 앞에서 설명한 수신인주소를 재작성할 때 사용되는 Rule-set부분이 정의된 부분이다.

```
Mlocal, P=/usr/bin/procmail, F=lsDFMAw5:/|@qSPfhn9,
S=EnvFromL/HdrFromL, R=EnvToL/HdrToL,
        T=DNS/RFC822/X-Unix,
        A=procmail -t -Y -a $h -d $u
```

그 다음 부분은 이 전자우편프로그람에서 사용되는 프로그람에 대한 경로지정에 대한 설정부분이다.

표 5-6은 Flags에 사용될수 있는 선택항목들이다.

표 5-6. Flags추가선택항목

선택항목	기 능
C	콤퓨터의 이름을 자동으로 첨가한다.
D	머리부에 date정보를 첨가한다.
F	머리부에 form:부분을 첨가한다.
M	머리부에 Message-ID부분을 첨가한다.
X	마침표로 시작하는 통보문에서는 여분의 마침표를 요구한다.
u	사용자이름의 대소문자를 구별한다.
a	ESMTP로 통보문교환한다.
w	/etc/passwd에 등록된 사용자를 확인한다.
m	한번에 여러 사용자에게 우편을 보낸다.

```
F=lsDFMAw5
```

우의 부분의 DFM은 각각 Data,Form,Message-ID머리부가 필요하다는것을 의미한다. 그리고 w는 /etc/passwd에 등록된 사용자를 확인하라는 의미이다. 만약 /etc/passwd 파일이 아닌 자체의 자료기지를 가지고 사용자를 관리할 경우에는 w를 삭제하면 된다.

```
A=procmail -Y -a $h -d $u
```

우의 부분은 procmail에 대한 추가선택을 설정한 부분이다. -Y는 전자우편합의형식을 규정한 부분이다. -a $h는 -a 다음에 오는 인수를 procmail인수로 받아들인다는것을 의미한다. 우의 설정에서는 sendmail이 받아들인 $h변수를 procmail변수로 사용하도록 설정하였다.

마지막으로 -d $u는 -d설정으로 delivery mode를 설정하고 $u라는 주콤퓨터변수에 저장된 값으로 전자우편을 전송하라는 설정이다.

여기서 $u는 sendmail에서 Rule set 0에 의해서 정의되여진 수신자를 의미한다. -d 추가선택을 사용하기 위해서는 procmail이 setuid로 설정되여있어야 한다.

```
Msmtp,       P=[IPC],                                   F=mDFMuX,
         S=EnvFromSMTP/HdrFromSMTP,           R=EnvToSMTP,
         E=\r\n, L=990,
             T=DNS/RFC822/SMTP,
             A=TCP $h
Mesmtp,      P=[IPC],                                   F=mDFMuXa,
         S=EnvFromSMTP/HdrFromSMTP,           R=EnvToSMTP,
         E=\r\n, L=990,
             T=DNS/RFC822/SMTP,
             A=TCP $h
Msmtp8,      P=[IPC],                                   F=mDFMuX8,
         S=EnvFromSMTP/HdrFromSMTP,           R=EnvToSMTP,
         E=\r\n, L=990,
             T=DNS/RFC822/SMTP,
             A=TCP $h
Mrelay,      P=[IPC],                                   F=mDFMuXa8,
         S=EnvFromSMTP/HdrFromSMTP,           R=MasqSMTP,
         E=\r\n, L=2040,
             T=DNS/RFC822/SMTP,
             A=TCP $h
```

우의 부분들은 TCP/IP우편을 배달하는데 사용되는 전자우편프로그람을 정의하는 부분이다.

여기서는 여러가지의 전자우편프로그람을 정의하고 있다. 우선 전통적인 ASCII SMTP 우편을 전송하도록 설계된 smtp와 Extended SMTP인 esmtp이다. Smtp8전자우편프로그람은 8비트자료를 처리한다. Relay는 TCP/IP우편을 외부우편중계봉사기를 통해서 중계를 하는 전자우편프로그람이다.

이러한 전자우편프로그람의 구조는 모두 같기때문에 대표적으로 smtp에 대해서만 보기로 하자.

실례에서 볼수 있는것처럼 전자우편프로그람정의를 의미하는 M지령으로부터 시작된다.

```
P=[IPC]
```

우의 부분은 전자우편프로그람에 사용되는 프로그람이 위치하고있는곳을 정의한다. 이 부분에서는 [IPC]라는 경로를 사용하였다. 이것은 Sendmail이 이 우편을 배달한다는것을 의미한다.

```
F=mDFMuX
```

우의 부분은 앞에서도 보았듯이 Sendmail의 기발을 정의하는 부분이다.(표 5-6을 참조) 이 부분에서 m은 해당 전자우편프로그람이 한번에 여러명의 수신자에게 전자우편을 보낼수 있다는것을 의미한다. DFM은 각각 Date, Form, message-ID머리부를 추가한다는것을 나타낸다. u는 주콤퓨터와 사용자이름에서 대소문자를 구별한다는것을 나타낸다.

S=11/31

이 부분은 앞에서 설명한 주소의 재작성규칙중 발신자의 주소를 재작성할 때 쓰이는 S (Rule set)에 대한 설정이다. 여기서 설정된 규칙수자는 전자우편프로그람마다 다르다. 이 부분에서는 우편에 명시된 발신인주소를 rule set 11을 리용해서 전자우편주소를 재작성하고 통보문에 명시된 발신인의 주소는 rule set 31을 통해서 재작성하겠다고 정의한것이다.

R=21

이 부분도 앞에서 알아본 주소재작성규칙들중 수신자의 주소를 재작성할 때 쓰이는 R Rule set에 대한 설정이다. 여기에서 사용된 수자는 전자우편프로그람마다 다르기 때문에 서로 다른 방법으로 주소를 처리하여야 한다. 여기서는 해당 전자우편프로그람이 모든 수신인 주소를 Rule set 21을 통해서 주소를 처리하겠다고 정의한것이다.

E=/r/n

이 부분은 전자우편이나 통보문에서 사용할수 있는 매개의 행을 어떤 규칙으로 끊을것인지에 대한 설정이다. 여기서는 매개의 행을 복귀(carriage return)와 행바꾸기로 끊었다.

L=990

이 부분은 전자우편이나 통보문에서 사용할수 있는 통보문행의 최대 길이에 대한 설정이다. 여기서는 통보문행의 최대길이를 990byte로 설정하였다. 다시말하여 최대길이가 990byte인 통보문들을 전송할수 있다.

T=DNS/RFC822/SMTP

이 부분은 해당 전자우편프로그람이 처리하는 통보문의 류형을 MIME형으로 설정한것이다. 또한 전자우편프로그람은 DNS주콤퓨터명과 RFC822형식에 의거한 전자우편주소와 SMTP오유코드를 사용할수 있도록 설정되여있다.

지금까지 sendmail.cf파일의 설정에 대하여 고찰하였다. Sendmail.cf파일은 그 중

요성에 비하여 볼때 관리자가 직접 설정하여야 할 부분이 거의 없다. 그것은 m4마크로를 리용하여 sendmail.cf파일이 작성되였기 때문에 해당 체계에 적절하게 설정되기 때문이다. 그러나 이 파일을 리해하는것이 곧 sendmail의 동작과 전자우편전송과정에 대한 옳은 인식을 가지는것이므로 이 파일에 대하여 잘 알고 있어야 한다.

제4절. 봉사기관리도구를 리용한 전자우편봉사기의 설정과 기동

앞에서 설명한 여러 봉사기들과 같이 Sendmail봉사기도 봉사기관리도구를 리용한 설정과 기동을 진행할수 있다.

이 절에서는 봉사기관리도구를 리용한 Sendmail봉사기의 설정과 기동에 대하여 보도록 하자.

봉사기관리도구의 대면부에서 《sendmail구성》항목을 찰칵하면 그림 5-12와 같이 Sendmail봉사기를 설정하는 대면부가 나타나게 된다.

매개 항목들을 선택하여 자기가 사용하려는 봉사기의 특성에 맞게 설정을 진행하면 된다. 봉사기의 설정을 다 하였으면 봉사기를 기동시키면 된다.

그림 5-12. Sendmail봉사기의 설정대면부

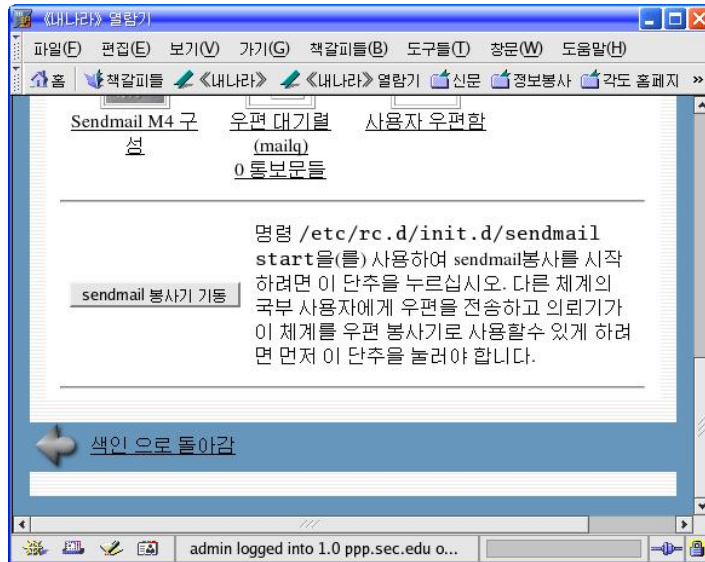



그림 5-13. Sendmail봉사기의 기동대면부

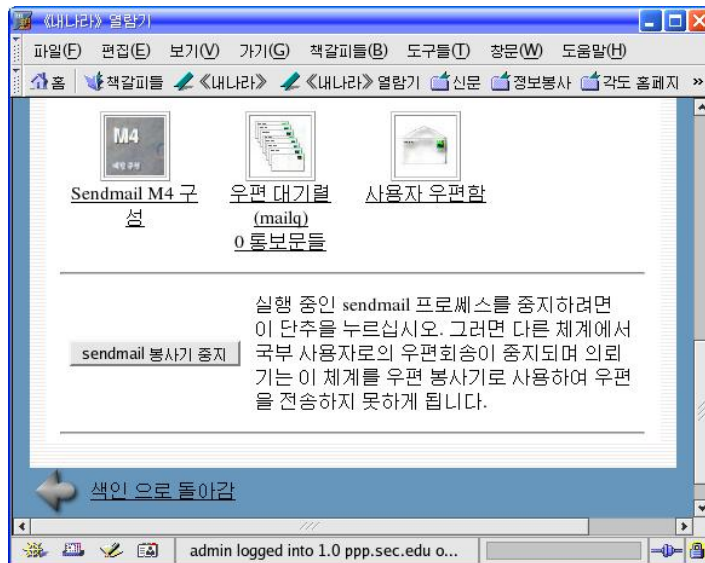



그림 5-14. Sendmail봉사기의 중지

봉사기의 기동은 우의 그림과 같이 대면부에 나오는 《Sendmail봉사기 기동》항목을 찰칵하면 된다.

봉사기가 정확히 기동하면 대면부의 이 부분이 그림 5-14와 같이 바뀌게 된다. 봉사기의 설정이 제대로 되지 않거나 봉사기의 기동시 오유가 발생하였으면 그림 5-14와 같은 대면부가 나오지 않고 오유통보문이 나오게 된다.

Sendmail봉사기를 중지하여야 할 필요가 있을 경우(항목 내용의 추가와 변경 등)에는 그림 5-14와 같은 대면부에서 《Sendmail봉사기 중지》를 찰칵하면 된다.

제5절. 전자우편의 주고받기

앞에서 설명한대로 설정과 기동을 진행하였으면 전자우편봉사기의 설정이 끝나고 전자우편봉사기가 기동하고 있다고 볼수 있다. 만일 전자우편봉사기가 제대로 설정되였다면 의뢰기에서 전자우편의 주고받기가 제대로 되여야 한다.

이 절에서는 사람들속에서 널리 리용되고있는 조작체계인 Windows조작체계를 사용한 의뢰기에서 전자우편봉사를 어떻게 주고받는가에 대하여 기본을 두고 설명을 진행한다.

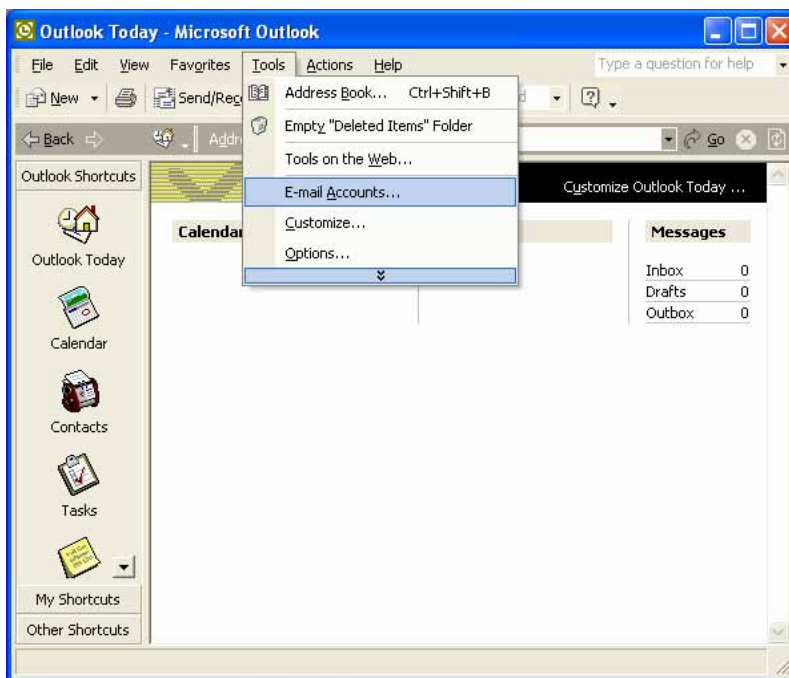



그림 5-15. Microsoft outlook의 시작대면부

Windows체계에는 표준적으로 제공하는 전자우편의뢰프로그람으로서 《Microsoft outlook》라는 프로그람이 있다. 조작체계의 판본에 따라서 약간씩 다르지만 대체로 같으므로 여기에서는 WindowsXP조작체계를 기본으로 놓고 설명을 진행한다.

《Start》를 찰칵하고 제일 웃쪽에 있는 《Microsoft outlook》항목을 찰칵한다. 그러면 그림 5-15와 같은 대면부가 펼쳐진다.

다음 Microsoft outlook의 대면부에서 Tools내림차림표를 찰칵하고 《E-mail Accounts》항목을 찰칵하면 그림 5-16과 같은 대면부가 펼쳐진다.

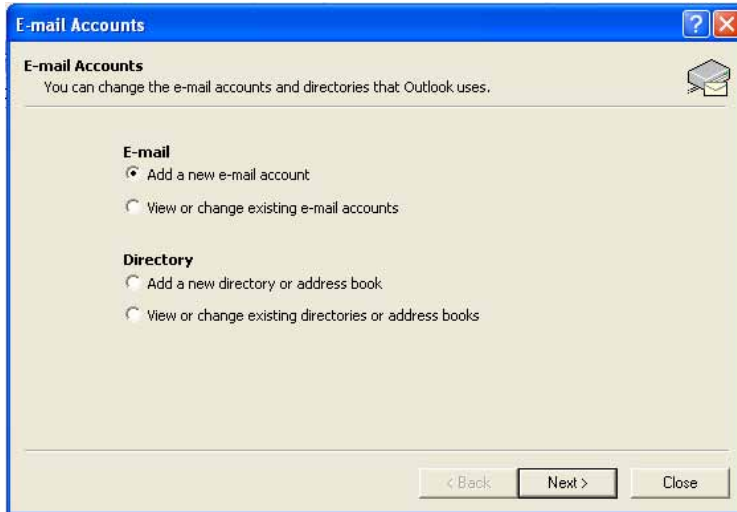



그림 5-16. E-mail Accounts설정대면부

그림 5-16과 같은 대면부가 펼쳐지면 새로운 전자우편사용자등록자리를 추가하기 위하여 《Add a new e-mail account》를 선택하고 《Next》단추를 찰칵한다.

그러면 그림 5-17과 같이 어떤 봉사기로부터 우편물을 받겠는가를 선택하는 항목이 나온다. 해당한 봉사기를 선택한 다음 《Next》단추를 찰칵하면 해당한 사용자정보와 봉사기의 정보를 요구하는 대면부가 나타난다. 여기에 필요한 설정값들을 넣어주면 전자우편을 받을 준비가 다 된다.

이러한 설정과정을 마치면 Microsoft Outlook파일을 리용하여 전자우편을 보내거나 받을수 있다.

그림 5-17. 봉사기류형의 설정

그림 5-18. 사용자정보 및 봉사기정보의 설정

제6장. Samba봉사기

Linux조작체계에 숙련된 사용자라고 하더라도 아직까지 많은 응용프로그람이 Windows조작체계에서만 동작하도록 되여있기때문에 Linux조작체계만을 사용하지 않고있는것이 현재의 실태이다. 응용프로그람이 적은것이 Linux조작체계의 약점이라고도 할수 있다. 이러한 원인들로 하여 많은 사용자들이 Linux조작체계와 Windows조작체계를 모두 사용하고있는데 이렇게 사용하다보면 두가지의 OS를 공유하여 사용할 일이 생기게 된다. 하지만 서로 다른 조작체계이기때문에 공유하고 싶어도 일반적인 방법으로는 공유할수 없다.

이러한 문제를 해결해 준것이 Samba봉사기이다. 이 장에서는 Samba의 개념에 대해 설명을 하고 거기에 기초하여 실제적으로 봉사기를 구축하고 관리하는 방법에 대해 설명하였다. 그리고 Samba봉사기를 리용하여 서로 다른 OS에서 인쇄기를 사용할수 있게 하는 방법에 대해서도 설명을 진행한다.

이 장은 다음과 같은 체계로 구성되여있다.

- Samba봉사기의 개념과 구축
- Samba봉사기설정파일
- Windows로부터 공유등록부를 리용하는 방법
- 관리도구를 리용한 Samba봉사기의 관리

제1절. Samba봉사기의 개념과 구축

6.1.1. Samba봉사기의 간단한 개념

Linux조작체계를 많이 다루어 거기에 숙련된 사용자라고 하여도 Windows조작체계를 사용할 필요성에 대해 느낄 때가 있다. 또한 체계를 두가지이상 사용하여 Windows와 Linux조작체계와 같은 여러가지 OS를 같이 사용하고있다면 매개 조작체계의 구동기를 공유하게 된다.

Linux조작체계와 Windows조작체계는 서로 다른 조작체계이므로 필요할 때마다 매번 이 체계에서 저 체계에로 왔다갔다하는것은 매우 힘든 일이다.

Samba는 이러한 작업들을 쉽고 더 편안하게 하여 준다.

Samba는 Linux조작체계나 Windows98, WindowsNT, OS/2등 모든 조작체계와 인쇄기를 공유할수 있다. Linux조작체계의 령역을 Samba봉사기를 통해 공유해놓으면 Windows에서는 망환경에서 다른 Windows의 등록부를 공유하여 사용하는것처럼 똑같이 사용할수 있다. 또한 인쇄기가 하나밖에 없어 Linux조작체계에 련결해놓았다면 Windows조작체계사용자들은 표준상태에서 인쇄를 할수 없는데 이러한 문제도 Samba봉사기를 리용하여 해결할수 있다.

Samba봉사기(간단히 Samba)를 리용하면 Linux를 안정하게 Windows봉사기로써 사용할수 있다. 보통 Linux봉사기로써는 Windows조작체계의 자료를 취급할수 없다. 때문에 Windows와의 자료통신을 Samba를 통하여 실행하며 Samba는 Linux에 자료를 넘겨준다. 이처럼 Samba는 Linux와 Windows와의 《다리》의 역할을 수행하게 된다.

그외에도 인쇄기공유기능이나 망을 관리하는 주조종기의 기능 및 Windows봉사기를 대신하기 위한 기능도 가지고있다.

6.1.2. Samba봉사기의 설치와 그의 구조

먼저 봉사기의 설치에 대해서 보도록 하자.

봉사기의 설치는 앞에서도 본것처럼 RPM으로의 설치와 원천파일로부터의 설치가 있다. 그러면 이 가운데서 RPM으로의 설치에 대해서만 보기로 하자.

먼저 아래와 같은 지령을 사용하여 봉사기를 설치하려는 콤퓨터에 Samba봉사기가 설치되여 있는가, 없는가를 확인해본다.

```
#rpm -qa | grep samba
```

이 지령은 이미 앞에서 많이 사용한 지령이므로 이 지령의 의미와 그의 실행결과에 대해서는 구태여 설명하지 않겠다. 만일 봉사기가 설치되지 않았다는 결과가 나오면 아래와 같은 지령을 리용하여 RPM을 리용한 설치를 진행한다.

```
#rpm -Uvh samba*
```

설치가 제대로 진행되였다면 일단 데몬을 가동시켜 본다.

데몬을 가동시키면 smbd와 nmbd가 실행되는것을 확인할수 있다.

여기서 smbd는 Samba를 리용한 콤퓨터망을 통해 파일과 인쇄기를 공유하고 Samba의뢰기의 인증과 암호화를 제공하는 데몬이다. 그리고 nmbd는 WINS(Windows Internet Name Service)를 사용할수 있게 하고 열람을 도와주는 데몬이다.

보는바와 같이 Samba봉사기의 설치는 아주 간단하다.

그러면 봉사기의 등록부구조에 대해 보도록 하자. 그림 6-1에 Samba봉사기의 등록부구조를 보여주었다.

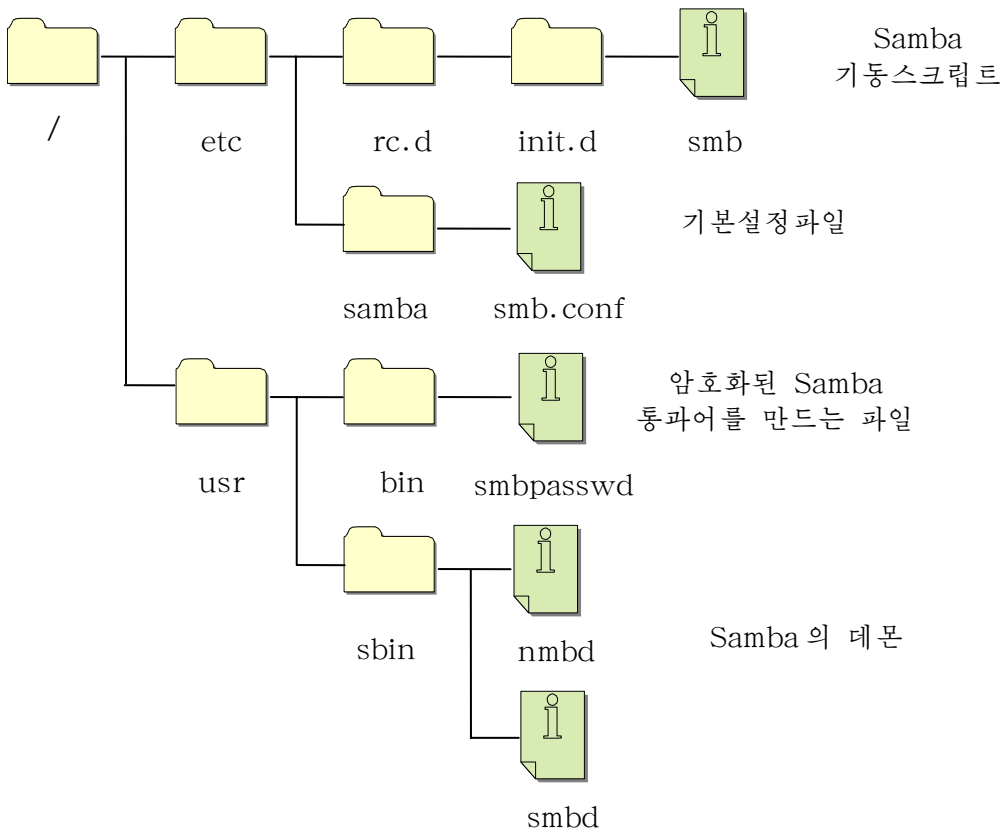



그림 6-1. Samba 의 등록부구조

그림에서 보는바와 같이 Samba봉사기는 많은 파일과 등록부들로 구성되여있다. 여기에서 기본으로 되는 파일은 smb.conf파일이다.

이 파일의 경로와 이름은 /etc/samba/smb.conf이다.

이 파일에서는 Samba봉사기에 대한 자세한 설정을 진행한다. 그러므로 Samba봉사기에서는 이 파일이 기본핵심파일이라고 말할수 있다. 그러므로 먼저 smb.conf파일이 있는지 확인하여 보아야 한다.

smb.conf의 구성과 설정방법에 대해서는 뒤에서 자세히 설명하므로 여기서는 기본적인 줄거리에 대해서만 보기로 하자.

이 파일은 [global] [homes] [printers]의 3개 부분과 공유등록부를 설정하는 부분으로 이루어져 있다.

[global]부분은 Samba의 설정에 대하여 설정한 부분이다.

[homes]부분은 사용자가 자기의 Home등록부에 대한 설정을 진행한 부분이다

[printers]부분은 공유인쇄기에 대한 설정을 진행한 부분이다.

6.1.3. 완전접근공유등록부의 만들기

여기서 제일 중요한 문제로 되는것은 완전접근공유등록부의 작성이다.

모든 사용자가 완전히 접근가능한 등록부를 만들기위해서는 다음과 같은 절차를 지켜야 한다.

- 암호인증을 진행하지 말아야 한다.
- 모든 사용자의 접속을 단일한 손님사용자로 취급하여야 한다.

그러면 한번 실지로 공유하려는 등록부를 하나 만들어보자.

/home등록부에 있는 share1이라는 등록부를 완전공유등록부로 만들어보자.

먼저 share1등록부가 있는가 보고 없으면 다음과 같이 만든다.

```
# mkdir/home/share1
```

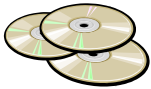
다음 아래와 같이 이 등록부의 허가권한을 변경한다. 즉 share1등록부를 완전조종이 가능하도록 만든다.

```
chmod 777 /home/share1
```

이렇게 한 다음 smb.conf파일을 편집한다.

smb.conf파일은 주석행이 많기 때문에 파일의 편집시 주의하여 편집을 하여야 한다.먼저 아래와 같은 지령을 사용하여 설치시 만들어진 기본파일의 여벌을 만든다. 왜 그렇게 해야 하는가는 구태여 언급하지 않는다.

```
# mv /etc/samba/smb.conf  /etc/samba/smb.old
```

다음 편집기를 리용하여 smb.conf파일을 편집한다.

실례 6-1: smb.conf파일의 일부

[global]
security=share ← security방식의 설정을 share로 한다.
coding system=euc ← 문자코드를 euc로 리용한다.
client codepage=949← Windows 영문판인경우 경우 949로 지정한다.
workgroup=it boost←작업그룹명이다. 의뢰기콤퓨터와 같은것을 리용한다.
[public]
path=/home/share1 ← 공유하려는 등록부를 지정한다.
writeable=yes ← 쓰기유효
guest ok=yes ← 손님사용자 리용을 허가한다.
guest only=yes ← 손님사용자가 리용한다.

우와 같이 파일에 수정을 진행한후 Samba를 실행시킨다.
봉사기를 실행시키려면 아래와 같은 지령을 주면 된다.

```
#/etc/rc.d/init.d/smb start
```

이 설정에 의해 전체 사용자로부터의 접근은 nobody사용자(기정손님사용자)로 바꾸어지게 된다.

여기서 문자코드를 euc로 한것은 Linux에서부터 파일이름을 표준으로 취급하여 Windows상에 표준으로 표시되도록 하기 위해서이다. 누구나 리용할수있기때문에 새로운 사용자 추가가 필요하지 않다는 우점이 있으나 파일의 루설, 분실 등의 오유가 생길 위험이 있다.

6.1.4. 호출권이 붙은 등록부의 작성

다음으로 특정하게 허가된 사용자만이 리용할수 있는 등록부를 작성하여 보자.
이를 위해서는 다음과 같은 2가지 단계를 거쳐야 한다.

- Samba사용자를 작성한다.
- 사용자인증을 진행한다.

Linux상에 작성되여 있는 기존의 사용자가 아닌 새로운 사용자를 등록하기 위해서는 아래와 같은 지령을 리용하여 Linux체계에 사용자를 만들고 그만이 사용하는 통과암호를 설정하여야 한다.

```
#useradd user2
#passwd user2
```

이런 과정을 거친 다음 Samba에 통과암호를 설정한다.

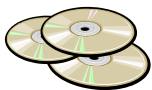
```
#smbpasswd -a user2

new smbpasswd:
add user user2
```

이 Samba통과암호는 Linux에서의 사용자등록가입통과암호와 같지않아도 되지만 편리를 위하여 동일한것을 사용할수도 있다. Linux에서처럼 사용자의 통과암호작성지령을 나누어 주지 않아도 Samba우에서 사용자를 작성할 때 smbpasswd 다음에 -a의 인수를 써서 실행시킬수 있다. 인수를 붙이지 않으면 통과암호를 변경하는 지령이 된다.

아래의 지령을 리용하면 Samba봉사기가 Linux에서 작성된 사용자를 일괄처리 한다.

```
#mksmbpasswd.sh /etc/passwd /etc/samba/smbpasswd
```

이 조작은 Linux의 사용자정보(/etc/passwd)를 mksmbpasswd.sh지령으로서 Samba의 사용자정보(/etc/samba/smbpasswd)에 적용하는 과정이다.

우와 같이 지령들을 실행시킨 다음에는 smb.conf파일을 다음과 같이 편집한다.

실례 6-2: smb.conf파일의 일부

[global]	
security=user	security방식의 설정을 user로 한다.
coding system=euc	← 문자코드를 euc로 리용한다.
client code page=932	← Windows일문판을 사용하는 경우 932로 지정한다.
workgroup=it boost	←작업그룹명. 의뢰기콤퓨터와 동일한것을 리용한다.
encrypt passwords=yes	← Windows의 암호화된 통과암호의 인증이

필요하다.

```
create mask=0777          ←파일작성시 증가속성을 무효로 설정
directory mask=0777       ←등록부작성시 허가속성을 무효로 설정
[public]                  ← 공유등록부의 이름이 된다.
path=/home/share1         ← 공유하려는 등록부를 지정한다.
writeable=yes             ← 쓰기유효
guest ok=no               ← 손님사용등록자리용을 허가하지 않는다.
```

우와 같이 설정을 진행하면 Samba봉사기상에 등록되여 있는 사용자이외에는 봉사기를 호출할수 없다.

우와 같이 파일의 수정을 진행한 경우 변경내용을 적용하려면 smaba봉사기를 재기동하여야 한다. 재기동하는 지령은 아래와 같다.

```
#/etc/rc.d/init.d/smb restart
```

지금까지 Samba봉사기의 간단한 개념과 그의 설치 및 간단한 조작에 대해 보았다.

이제는 Samba봉사기의 구축파일들에 대하여 구체적으로 보도록 하자.

제2절. Samba봉사기설정파일

이 절에서 Samba봉사기의 상세한 설정파일에 대하여 취급한다. Samba의 기본설정파일들은 앞에서도 보았지만 /etc/samba에 있다.

여기서 먼저 보아야 할 파일은 앞에서도 언급하였지만 smb.conf파일이다.

이 파일에서 반두점(;)과 #문자는 주석을 의미한다. 그러므로 이러한 문자의 뒤에 놓이는 문장들은 실지 봉사기의 기동에 아무런 영향도 미치지 못한다. smb.conf파일은 여러개의 부분들로 구성되여있다.

그러면 매개 부분의 거기에서 사용되는 추가선택항목들에 대하여 보기로 하자.

6.2.1. smb.conf파일

smb.conf파일은 아래와 같은 부분들로 이루어져있다.

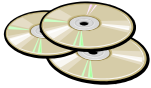
전역설정부분(global):이 부분에 포함된 내용은 Samba 봉사기전체설정에 영향을 준다.

공유정의부분(homes): 의뢰기가 특별한 ID로 접근할 때 봉사기의ID와 열쇠단어가 일치하면 다른 홈등록부에 대한 설정이 없어도 ID에 대한 홈등록부를 련결할수 있다.

인쇄기정의(printers)부분: 이 부분은 주콤퓨터의 printcap파일을 읽어 인쇄기를 련결할수 있게 하여준다.

그러면 smb.conf파일에 대해 구체적으로 보도록 하자.

아래에 smb.conf파일의 간단한 실례를 주었다.

- smb.conf파일

실례 6-3: smb.conf파일

```
#
#======================= 전역설정=======================
[global]

# workgroup = NT-Domain-Name or Workgroup-Name
;  workgroup = MYGROUP
```

server string 은 NT 서술마당과 맞먹는다.

```
;  server string = Samba Server
```

다음의 추가선택은 보안을 위해서 중요한 추가선택이다. 이것은 국부망안의 콤퓨터에로의 련결을 제한할수 있게 한다.

다음의 실례는 두개의 C클라스망에로의 접근만을 허용한다. 구체적인 문법실례는 smb.conf man page 를 보면 된다.

```
;   hosts allow = 192.168.1. 192.168.2. 127.
```

만일 필요한 인쇄기를 개별적으로 설치하기전에 자동적으로 설치되게 하려면 다음과 같이 한다.

```
   printcap name = /etc/printcap
   load printers = yes
```

이것은 인쇄기가 표준이 아닌 경우에는 필요없는 추가선택이다.

현재 지원하고있는 인쇄체계는 다음과 같다.

즉

```
   bsd, sysv, plp, lprng, aix, hpux, qnx
   printing = lprng
```

만일 손님사용자등록자리을 바란다면 여기서 주석을 제거하고 /etc/passwd에 이것을 추가하여야 하며 다른 경우에는 사용자 “nobody”로 사용된다.

; guest account = pcguest

이것은 Samba에 련결된 매 콤퓨터에 따라서 일지기록파일을 구별하여 사용하도록 Samba에 통지한다.

log file = /var/log/samba/%m.log

일지기록파일의 최대크기를 kb 로 설정한다.

max log size = 0

보안방식이다. 많은 사람들이 사용자준위보안을 원할것이다. 구체적인것은 security_level.txt 에 있다.

security = user

password server 추가선택은 오직 security = server와 함께 사용하여야 한다. 인수는 다음과 같은것을 포함할수 있다.

password server = My_PDC_Name [My_BDC_Name] [My_Next_BDC_Name]

or to auto-locate the domain controller/s

password server = *

; password server = <NT-Server-Name>

통과암호준위는 통과암호문자렬의 문자수와 대소문자를 구별하는가를 규정한다.

; password level = 8

; username level = 8

통과암호를 암호화 할것을 바란다면 이것을 설정하면 된다. 설정하려면 samaba문서의 ENCRYPTION.txt, Win95.txt 와 WinNT.txt 를 보면 된다. 이 문서들을 보지 않고서는 이 추가선택을 설정할수 없다.

encrypt passwords = yes

smb passwd file = /etc/samba/smbpasswd

다음의 추가선택은 Samba가 SSL을 함께 지원하도록 건설되였을 때 가짜오유의 발생으로부터 smbclient를 보존하는데 필요하다.

; ssl CA certFile = /usr/share/ssl/certs/ca-bundle.crt

다음의 추가선택은 Windows에서 Linux조작체계에로 통과암호를 변경시키는것을 허락할 때 필요하다.

[주해1] : 이것은 'encrypt passwords' 와'smb passwd file' 우에서 사용된다.

［주해2］:작업기에서 암호화된 smb통과암호만을 변경시킬것을 허가하는데는 필요하지 않다. 그것들은 smb통과암호가 syn로써 유지된 unix통과암호를 허락한다.

unix password sync = Yes
passwd program = /usr/bin/passwd %u
passwd chat = *New*password* %n\n *Retype*new*password* %n\n
*passwd:*all*authentication*tokens*updated*successfully*

pam통과암호변경을 허락한다.

pam password change = yes

Unix 사용자들은 각이한 SMB사용자이름들로 넘길수있다.

; username map = /etc/samba/smbusers

다음의 행을 리용하여 자기의 설정을 매 콤퓨터마다에서 전용화할수 있다. %m은 사용자의 콤퓨터에 련결된 망기본입출력체계로 교체된다.

; include = /etc/samba/smb.conf.%m

이 파라메터는 pam등록자리과 대화접속관리지령들을 따를것인가 말것인가를 지정한다.

표준동작은 본문인증만을 지우기 위하여 pam을 리용하고 다른 등록자리와의 대화접속관리지령을 무시한다.

Samba가 encrypt passwords = yes 인 경우 pam인증은 무시된다.

obey pam restrictions = yes

자세한것은 speed.txt 와 manual pages를 보면 된다.

socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192

Samba가 다중대면부를 리용하도록 설정한다.

만일 다중망대면부를 가지고있다면 여기에 그것들을 입력하여야 한다.

; interfaces = 192.168.12.2/24 192.168.13.2/24

원격열람목록동기를 여기서 설정할수 있다.

특별한 주콤퓨터나 혹은 자신을 전체 부분망에 알리는것을 설정할수 있다.

; remote browse sync = 192.168.3.25 192.168.5.255

; remote announce = 192.168.1.255 192.168.2.44

열람기조종추가선택:

만일 Samba가 국부마스터로 되기를 원하지 않을때 설정할수 있다. 다른 경우에는 표준선택규칙을 적용한다.

; local master = no

#OS Level은 주열람기선택에서 이 봉사기의 순서를 결정한다.
#기정값은 합리적으로 되게 할것이다.
;   os level = 33

Samba가 령역주열람기로 되게한다.이것은 Samba가 부분망사이에서 열람목록을 비교하도록 허락한다. 만일 Windows NT 령역조종기가 항상 이 작업을 하고있다면 이것을 쓰지 말아야 한다.
;   domain master = yes

Preferred Master는 Samba가 기동시에 강제적으로 국부열람기로 되도록 한다.
;   preferred master = yes

Windows95콤퓨터에서 Samba가 령역등록가입봉사기가 될것을 원한다면 이것을 리용할수 있다.
;   domain logons = yes

만일 령역등록가입이 가능하다면 매 콤퓨터와 매 사용자등록가입스크립트가 특수한 등록가입묶음파일이 매 콤퓨터에서 실행되도록 할수 있다.
;   logon script = %m.bat

매사용자마다 특수한 등록가입묶음파일의 실행
;   logon script = %U.bat

이동프로파일을 저장하는 곳(Win95 와 WinNT에서만) %L 이 봉사기의 망기본입출력체계이름으로 치환, %U 사용자이름으로 치환한다. [Profiles] 에서 아래와 같은 추가선택의 주석(;)을 제거하여야 한다.
;   logon path = \\%L\Profiles\%U

Windows Internet Name Serving Support Section:
WINS Support - Samba의 NMBD 부분에 wins server가 가능하다는것을 통지한다
;   wins support = yes

WINS Server - Samba의 NMBD부분에 wins client가 가능하다는것을 통지한다. Samba는 WINS Server 혹은 WINS Client중에서 어느 하나만이 될수 있다.
;   wins server = w.x.y.z

WINS Proxy - Samba가 망안에 유능한 wins client가 없을 때 이름요구해결에 대답하기 위한것이다. 이를 위해서는 망안에 적어도 하나의WINS Server가 있어야 한다. 기정값은 no 이다.
;   wins proxy = yes

DNS Proxy - DNS nslookups에 의해서 NetBIOS이름을 해결하겠는가 아닌가를 Samba에 알려준다.기정값은 no이다.

```
 dns proxy = no

# 보존하는것이 편리한 경우 이 추가선택을 쓴다. 기정값은 no 이다.
;  preserve case = no
;  short preserve case = no

# 기정값으로 모든 dos파일을 표준큰글자로 한다.
;  default case = lower

# 대소문자구별에 주의를 돌려야 할 때 설정한다.
;  case sensitive = no

#============================공유정의부분=====================
[homes]
   comment = Home Directories
   browseable = no
   writable = yes
   valid users = %S
   create mode = 0664
   directory mode = 0775
# 만일 Samba사용자들이 손님사용자로 넘겨진것을 인식하지 못할 때 쓴다.
; map to guest = bad user

# 다음부분에서 주석을 제거하면 령역등록가입을 위한 netlogon등록부를 만든다.
; [netlogon]
;   comment = Network Logon Service
;   path = /usr/local/samba/lib/netlogon
;   guest ok = yes
;   writable = no
;   share modes = no

주석을 제거하면 특별한 이동프로파일공유를 준비한다.기정으로는 홈등록부이다.
;[Profiles]
;    path = /usr/local/samba/profiles
;    browseable = no
;    guest ok = yes
```

[주해] 만일 BSD-style인 인쇄기를 가지고있으면 특별히 매개 인쇄기를 정의할 필요가 없다.

```
 [printers]
   comment = All Printers
```

```
   path = /var/spool/samba
   browseable = no
Set public = yes to allow user 'guest account' to print
   guest ok = no
   writable = no
   printable = yes
```

이것은 파일을 공유하는데 필요한것이다.

```
;[tmp]
;   comment = Temporary file space
;   path = /tmp
;   read only = no
;   public = yes
```

이것은 공개접근가능한 등록부인데 staff그룹만이 쓰기가능하고 다른 사용자들은 읽기만 할수 있다.

```
;[public]
;   comment = Public Stuff
;   path = /home/samba
;   public = yes
;   writable = yes
;   printable = no
;   write list = @staff
```

개인인쇄기로써 오직 김철수만이 사용가능하고 스풀자료는 김철수의 홈등록부로 치환된다.이 경우 김철수는 스풀등록부에 쓰기접근을 하여야 한다.

```
;[kimcholsuprn]
;   comment = Kimcholsu's Printer
;   valid users = kimcholsu
;   path = /home/kimcholsu
;   printer = kimcholsus_printer
;   public = no
;   writable = no
;   printable = yes
```

개인적인 등록부로써 오직 김철수만이 사용가능하다. 김철수가 쓰기접근을 요구하여야 한다.

```
;[kimcholsusdir]
;   comment = Kimcholsu's Service
;   path = /usr/somewhere/private
;   valid users = kimcholsu
;   public = no
;   writable = yes
```

;　printable = no

각이한 등록부를 각이한 콤퓨터가 접속할수 있게 봉사를 진행한다. 여기서 %u 추가선택은 사용자이름으로 바꾸고 %m은 련결된 콤퓨터이름으로 바꾸어야 한다.

;[pchome]
; comment = PC Directories
; path = /usr/local/pc/%m
; public = no
; writable = yes

공개적으로 접근가능한 등록부로써 모든 사용자에게 읽기/쓰기가 가능하다. 사용자의 등록부에 만들어진 모든 파일은 기정사용자에게 속하며 임의의 사용자가 다른 사용자의 파일을 지울수 있다. 명백하게 이 등록부는 표준사용자에 의해 쓰기가 가능해야 한다. 다른 사용자도 역시 그 파일소유자대신에 그 파일에 써넣을수 있다.

;[public]
;　path = /usr/somewhere/else/public
;　public = yes
;　only guest = yes
;　writable = yes
;　printable = no

다음의 실례는 두 사용자가 파일들을 특수한 사용자에 의해 소유될 곳에 놓을수 있도록 하기 위해서 등록부를 어떻게 공유하는가를 보여준다. 이 설정은 파일의 람용을 막고 두 사용자만이 써넣을수 있게 한다. 이 실례는 이것을 요구하는 많은 사용자들에게로 확장할수 있다.

;[myshare]
;　comment =Liyonghui's and Kimcholsu's stuff
;　path = /usr/somewhere/shared
;　valid users =Liyonghui kimcholsu
;　public = no
;　writable = yes
;　printable = no
;　create mask = 0765

- 설정파일에 대한 설명

우에서 보다싶이 smb.conf파일은 매우 복잡하게 구성되여있다. 또한 여기에 사용되는 추가선택항목들도 많다고 볼수 있다. 이 부분에서는 파일을 이루는 매 구성요소에 대한

구체적인 설명을 진행한다.

만일 추가선택항목이 전역설정부분에서만 리용된다면 [global]이 그 이름앞에 놓인다. 어떤 항목이라도 공백에 의하여 구별되며 주석은 제외하였다.

그러면 매개 항목에 대해 보기로 하자.

```
Admin users =user list
```

Samba에 의하여 뿌리권한이 부여된 사용자목록. 기정으로는 null

```
Allow hosts=host list
```

공유에 접근할수 있는 콤퓨터목록

```
available=Boolean
```

만일 no라고 설정하면 공유접근이 거부된다.

```
[global] bind interfaces only=Boolean
```

만일 yes라고 설정하면 대면부목록에서만 열람이 가능하게 된다.

```
Browseable= Boolean
```

열람목록에 대한 공유가 허가된다는것을 알린다.

```
[global] browse list= Boolean
```

가능한 값은 yes no

이 봉사기의 열람목록을 on/off 로 한다.

```
Character set =name
```

만일 설정을 하면 dos code page를 서유럽, 동유럽, 로씨아끼릴문자 등의 문자모임으로 번역한다. 의뢰기 code page는 850으로 되여야 한다.

```
Client code page=name
```

가능한 값은 다음과 같다.

- Code Page 437 - MS-DOS Latin US
- Code Page 737 - Windows '95 Greek
- Code Page 850 - MS-DOS Latin 1

- Code Page 852 - MS-DOS Latin 2
- Code Page 861 - MS-DOS Icelandic
- Code Page 866 - MS-DOS Cyrillic
- Code Page 932 - MS-DOS Japanese SJIS
- Code Page 936 - MS-DOS Simplified Chinese
- Code Page 950 - MS-DOS Traditional

Dos code page를 정확히 설정한후에 유효한 문자설정을 한다. 례로 850은 유럽이고 932는 일본어 shift-jis이다.

Coding system=code

가능한 값은 euc,cap,hex 등이 있다. 기정값은 null이다.

코드체계를 kanji로 하기 위해서는 client code page를 932로 설정하여야 한다.

SJIS - Shift-JIS. Does no conversion of the incoming filename.

JIS8, J8BB, J8BH, J8@B, J8@J, J8@H - Convert from incoming Shift-JIS to eight bit JIS code with different shift-in, shift out codes.

JIS7, J7BB, J7BH, J7@B, J7@J, J7@H - Convert from incoming Shift-JIS to seven bit JIS code with different shift-in, shift out codes.

JUNET, JUBB, JUBH, JU@B, JU@J, JU@H - Convert from incoming Shift-JIS to JUNET code with different shift-in, shift out codes.

EUC - Convert an incoming Shift-JIS character to EUC code.

HEX - Convert an incoming Shift-JIS character to a 3 byte hex representation, i.e. :AB.

CAP - Convert an incoming Shift-JIS character to the 3 byte hex representation used by the Columbia AppleTalk Program (CAP), i.e. :AB. This is used for compatibility between Samba and CAP.

Comment = text

가능한 값은 본문문자렬 혹은 null이다.

Microsoft directory window의 상세목록 혹은 net view에서 공유에 대한 주석을 설정한다. server string configuration추가선택을 참조하시오.

[global] config file=pathname

가능한 값은 unix pathname

현재Samba설정파일대신에 추가적인 설정파일을 읽도록 선택한다.

일부 사용자나 계산기를 위해서 사용자설정파일을 %변수를 리용하든가 혹은 설정파일을 재배치한다.

실례: config file = /usr/local/samba/lib/smb.conf.%m

```
Copy=section name
```

가능한 값은 존재하는 section이름

그것이 나타나는 공유에로 이전에 보였던 공유설정을 복사한다.

```
Create mask=octal value
```

가능한 값은 8진수로써 0-0777까지이다.

새로운 파일에 대한 최대가능한 허가를 설정한다.

기정: create mask = 0744

실례: create mask = 0775

```
Create mode = octal permission bits
```

동의어 create mask

```
[global] deadtime = minutes
```

적용되는 값은 분이다.

쓰이지 않는 련결을 끊을 때까지의 시간이다. 령은 무한대로 된다.

```
[global] debug level = number
```

적용되는 값은 수자이다.

Logging level사용을 설정한다.

```
[global] debug timestamp = boolean
```

가능한 값은 YES,NO이다.

Timestamp는 모든 일지통보문이다.그것이 유효하지 않을 때 꺼지게 된다.

```
[global] default = name
```

가능한 값은 공유이름이다.

```
default case = case
```

가능한 값은 LOWER,UPPER

새로운 파일이름을 저장할 때 설정한다.

[global] default service = share name

가능한 값은 공유이름이다.동의어는 default이다.

delete readonly = boolean

가능한 값은 no,yes이다.

읽기전용파일을 지울것을 요구할 때 지우는것을 허락한다.

이것은 dos/windows에서는 허가되지않는다. 그러나 등록부허가가 구별되는 보통 unix에서는 허가된다.

delete veto files = boolean

가능한 값은 no,yes

deny hosts = host list

가능한 값은 콤퓨터목록이다.

동의어는 hosts deny이다. 련결 혹은 공유가 거부된 기계의 목록을 서술한다.

[global] dfree command = command

여기에 사용되는 값은 shell지령이다. 지령은 봉사기에서 실행되며 디스크의 빈공간으로 되돌아가도록 봉사기에서 실행하는 지령이다.

directory = pathname

가능한 값은 경로이름이다.

보통 자동적으로 사용자의 home 등록부에 공유하기 위해서 [homes]안에 설정한다. 그렇지 않으면 기정으로 /tmp이다.

directory mask = 8진 허가비트수

directory mode = 8진 허가비트수

[global] dns proxy = boolean

가능한 값은 YES, NO 기정값은 YES

만일 YES로 설정하고 wins server = YES로 하면 사용중의 wins봉사기에서 그것들을 찾지못하면 주콤퓨터이름을 탐색한다.

```
[global] domain logons = boolean
```

가능한 값은 YES, NO 기정값은 NO

```
[global] domain master = boolean
```

가능한 값은 YES, NO 기정값은 no

이 추가선택을 yes로 하면 Samba로 하여금 령역주열람기가 되게 한다. 그러면 Samba는 부분망에서 열람목록을 모으는 작업을 한다. 하지만 이미 Windows의 NT령역조종기가 있거나 다른 Ssmaba봉사기가 master로 동작하고있으면 절대로 yes로 해서는 안된다.

```
dont descend = comma-list
```

가능한 값은 반점으로 구별된 경로의 목록. 기정값은 NULL

등록부서술에서 등록부변경과 탐색을 하지 못하게 한다. 이것은 열람편의추가선택이다. 이것은 그 어떤 외부보안도 하지 않는다.

```
dos filetimes = boolean
```

가능한 값은 YES, NO 기정값은 NO

만일 파일의 비소유자가 파일에 쓰기권한이 있다면 파일변경시간을 쓸수 있게 한다.

```
[global] encrypt passwords = boolean
```

가능한 값은 YES, NO 기정값은 NO

Windows NT형의 통과암호암호를 사용한다. Samba봉사기에서 smbpasswd를 요구한다.

```
exec = command
```

preexec 의 동의어로써 사용자가 공유에 련결하기전에 미리 지령을 실행한다.

```
fake directory create times = boolean
```

가능한 값은 YES, NO 기정값은 NO

Microsoft nmake사용자를 위해서 오유를 고정한다. 만일 설정하면 Samba는 nmake가 매 시각 모든 파일을 고칠수 없게 등록부를 설정한다.

```
force create mask = octal permission bits
```

가능한 값은 8진수로써 0 부터 0777 이다. 기정값은 0

```
force create mode = octal permission bits
```

가능한 값은 8진수로써 0 부터 0777 이다. 기정값은 0

force create mask와 같은 뜻이다.

force directory mask = octal permission bits

가능한 값은 8진수로써 0 부터 0777 이다.기정값은 0

force directory mode = octal permission bits

가능한 값은 octal value from 0 to 0777 기정값은 0

force group = unix group

가능한 값은 group 기정값은 NULL

모든 사용자가 공유에 접근할수 있게 그룹이름을 유효하게 할당하여 설정한다. 표준 사용자 그룹은 무시한다.

fstype = string

가능한 값은 NTFS, FAT, Samba의 기정값은 NTFS

파일체계형을 의뢰기에 보고하도록 설정한다.

[global] getwd cache = boolean

가능한 값은 YES, NO 기정값은 NO

실행을 위해서 현재등록부를 저장한다. wide links 추가선택과 함께 쓰는것이 좋다.

guest account = user

가능한 값은 username 기정값은 NULL

권한이 없는 unix등록자리이름을 공유인쇄와의 접근을 위해서 guest ok 를 함께 사용하여 설정한다.

guest ok = boolean

가능한 값은 YES, NO 기정값은 NO

만일 yes로 설정하면 통과암호가 이 공유에서 필요하지 않게 된다.

public와 동의어이다.

guest only = boolean

가능한 값은 YES, NO 기정값은 NO

강제적으로 공유사용자를 guest등록자리처럼 리용할수 있게 설정한다.

guest ok 와 public를 yes로 설정할것을 요구한다.

hide dot files = boolean

가능한 값은 YES, NO 기정값은 YES

만일 dos/windows에서 파일들이 숨은 속성으로 되였다면 그것들을 공유에서 점으로 시작된 파일로 취급한다.

hide files = slash-separated list

가능한 값은: /문자렬로 구별된 패턴의 목록. 기정값은 NULL

목록은 DOS의 파일속성에서 숨은 파일속성으로 설정하기 위한 파일이나 등록부이름의 목록. 이름은 ?,* 패턴문자렬과 %변수를 포함할수 있다.

invalid users = user list

가능한 값은 사용자목록. 기정값은 NULL

공유접근을 허가하지 않으려는 사용자목록

[global] max xmit = bytes

가능한 값은 size in bytes 기정값은 65535

Samba에 의하여 발생시키려는 파케트의 최대크기를 설정한다.

속도가 느린 련결과 낡은 의뢰기의 오유를 위해서 파라메터를 조절한다.

값은 2048보다 커야 한다.

[global] message command = /absolute_path/command

가능한 값은 shell command 기정값은 NULL

의뢰기에서 WinPopup 통보문이 도착했을 때 봉사기에서 지령을 설정한다.

min print space = kilobytes

가능한 값은 space in KB 기정값은 0 (unlimited)

인쇄기에 접근하기전에 요구되는 spool의 최소용량을 설정한다.

[global] min wins ttl = seconds

가능한 값은 time in seconds 기정값은 21600 (6 hrs)

nmbd WINS cashe에서 netbios의 time-to-live의 최소값시간을 설정한다.

name resolve order = list

가능한 값은 list of lmhosts, wins, hosts and bcast, 기정값은 lmhosts wins hosts bcast 이름으로부터 IP주소를 얻을 때 표탐색순서를 설정한다.

[global] netbios aliases = list

가능한 값은 list of netbios names, 기정값은 NULL
Samba봉사기가 자기자신을 광고하려는 보조적인 netbios이름을 추가.

```
netbios name = hostname
```

가능한 값은 host name, 기정값은 varies

```
[global] networkstation user login = boolean
```

가능한 값은 YES, NO, 기정값은 YES
만일 no로 설정하면 의뢰기는 security=server일 때 전혀 등록가입할수 없다.

```
[global] null passwords = boolean
```

가능한 값은 YES, NO, 기정값은 NO
만일 YES로 설정하면 빈통과암호를 가지고 등록자리에로의 접근을 허락한다.

```
only user = boolean
```

가능한 값은 YES, NO, 기정값은 NO
공유사용자가 username=list로 될것을 요구한다.

```
oplocks = boolean
```

가능한 값은 YES, NO, 기정값은 YES

```
[global] os level = number
```

가능한 값은 number, 기정값은 0
열람목록으로 선출되였을 때 봉사기의 후보를 설정한다. 만일 Samba가 다른 조작체계보다 높은준위로 Windows에 련결하려고 할 때 설정할수 있다. 준위는 작업그룹의 Windows와 Windows95에서는 1을 쓰고 Windows NT의뢰기는 17, 그리고 Windows NT봉사기는 33이다.

```
[global] packet size = bytes
```

가능한 값은 number in bytes 기정값은 65535

```
[global] passwd chat debug = boolean
```

가능한 값은: YES, NO 기정값은 NO
일지기록수준 100으로 전체 통과암호대화를 기록한다.

```
[global] passwd chat = command sequence
```

가능한 값은: Unix server commands 기정값은 compiled-in value

봉사기에서 통과암호를 변경하는데 쓰이는 지령을 설정한다.변수로써 %o(낡은 통과암호)그리고 %n(새 통과암호)다음으로 순서대로 \r \n \t그리고 \s(공백)를 쓸수있다.

```
[global] passwd program = program
```

가능한 값은: Unix server program 기정값은 NULL

사용자의 통과암호를 변경시키는데 쓰이는 지령을 설정한다.특권사용자권한으로 실행하여야한다. %u(사용자)를 지원한다.

```
[global] password level = number
```

가능한 값은: number 기정값은 0

통과암호를 정합하는데 쓰이는 대문자치환문자수를 서술한다

```
[global] password server = netbios names
```

가능한 값은: list of NetBIOS names 기정값은 NULL

통과암호가 유효하게 하려는 smb봉사기의 목록이다. NT통과암호봉사기와 security = server 그리고 security=domain설정추가선택과 함께 쓰인다. 이때 NT통과암호봉사기는 Samba봉사기로부터 등록가입이 가능하여야 한다.

```
panic action = /absolute_path/command
```

가능한 값은 완전히 알맞는 Unix쉘지령들이다. 기정값은 NULL

Samba가 파괴되였을 때 실행하려는 지령을 설정. Samba개발자와 검사자들에게는 /usr/bin/X11/xterm -display :0 -e gdb /samba/bin/smbd %d 가 가능한 값이다.

```
path = pathname
```

가능한 값은 pathname 기정값은 varies

파일과 인쇄기공유를 위한 등록부의 경로를 지정할 때 쓴다.자동적으로 [homes] 공유에서는 사용자의 홈등록부가 공유되며 다른 경우에는 표준으로 /tmp이다.

```
postexec = /absolute_ path/command
```

가능한 값은 fully-qualified Unix shell command 기정값은 NULL

공유가 끊어진후에 실행하려는 지령을 선택한다.

[참조] preexec, root preexec, and root postexec.

```
preexec = /absolute_path/command
```

가능한 값은 완전히 알맞는 Unix 쉘지령. 기정값은 NULL
공유에 련결하기전에 실행하려는 사용자지령을 설정한다.
[참조] postexec, root preexec, and root postexec.

```
[global] preferred master = boolean
```

가능한 값은 YES, NO 기정값은 NO
만일 yes로 설정하면 Samba는 우선권이 높은 기본열람기로 된다.

```
preload = share list
```

가능한 값은 list of services 기정값은 NULL
auto services 와 동의어이다. 열람목록에서 항상 나타나야 할 공유목록을 서술한다.

```
print command = /absolute_path/command
```

가능한 값은 완전히 알맞는 Unix 쉘지령.

```
print ok = boolean
```

가능한 값은 YES, NO 기정값은 NO
printable과 동의어이다.

```
printable = boolean
```

가능한 값은 YES, NO 기정값은 NO

```
[global] printcap name = pathname
```

가능한 값은 경로이름
[printers]공유에서 사용될 가능한 인쇄기파일을 설정한다.
기정값은 /etc/qconfig

```
printer = name
```

가능한 값은 printer name 기정값은 lp
unix인쇄기의 이름을 설정한다.

```
printer driver = printer driver name
```

가능한 값은 Windows에서 쓰이는 정확한 인쇄기구동프로그람이름.기정값은 NULL
주의: 값은 대소문자를 구별한다.

```
[global] printer driver file = path
```

가능한 값은 Unix pathname 기정값은 samba-lib/printers.def
Windows 95/98에서 사용가능한 msprint.def파일의 위치를 설정한다.

```
printer driver location = path
```

가능한 값은 Windows망경로.기정값은 \\server\PRINTER$
특별한 인쇄기를 위한 구동기의 위치를 설정한다.

```
printer name = name
```

가능한 값은 name 기정값은 NULL

```
printing = style
```

가능한 값은 bsd, sysv, hpux, aix, qnx, plp, lprng 기정값은 bsd
콤파일된 값대신에 그우에 하나의 인쇄양식을 설정한다.

```
[global] protocol = protocol
```

가능한 값은 NT1, LANMAN2, LANMAN1, COREPLUS, CORE 기정값은 NT1

```
public = boolean
```

가능한 값은 YES, NO 기정값은 NO
만일 YES로 하면 통과암호가 이 공유에서 필요없다. 동의어는 guest ok

```
queuepause command = /absolute_path/command
```

가능한 값은 유효한 Unix 지령. 기정값은 varies
인쇄를 정지시키는데 쓰이는 지령을 설정한다. 인쇄추가선택을 기정값으로 초기화한다.

```
queueresume command = /absolute_path/command
```

가능한 값은 valid Unix command 기정값은 varies

```
read bmpx = boolean
```

가능한 값은 YES, NO 기정값은 NO

```
read only = boolean
```

가능한 값은 YES, NO 기정값은 NO
공유를 읽기전용으로 설정한다.

```
[global] read prediction = boolean
```

가능한 값은 YES, NO 기정값은 NO

```
[global] read raw = boolean
```

가능한 값은 YES, NO 기정값은 YES

```
[global] read size = bytes
```

가능한 값은 size in bytes 기정값은 2048
망속도와 디스크가 정합되지 않을때 봉사기의 완충추가선택을 설정한다.

```
[global] remote announce = remote list
```

가능한 값은 list of remote addresses 기정값은 NULL
Samba봉사기가 자기자신을 알리기 위한 목록에 작업그룹을 추가한다.
IP주소/작업그룹으로 서술되는데 여러개의 그룹이면 공백으로 구별된다.(례로 192.168.8.15/os) 직접방송을 허락한다. 봉사기는 작업그룹의 열람목록에 나타나게 된다. WINS는 요구하지 않는다.

```
[global] remote browse sync = address list
```

가능한 값은 IP주소목록. 기정값은 NULL

```
[global] root = pathname
```

가능한 값은 Unix 경로. 기정값은 NULL

```
[global] root dir = pathname
```

가능한 값은 Unix 경로. 기정값은 NULL

```
[global] root directory = pathname
```

가능한 값은 Unix pathname 기정값은 NULL

```
root postexec = /절대경로/지령
```

가능한 값은 fully-qualified Unix shell command 기정값은 NULL
공유가 끊어진후에 특권사용자권한으로 실행하려는 지령을 설정한다.

```
root preexec = /absolute_path/command
```

가능한 값은: fully-qualified Unix shell command 기정값은 NULL
공유에 련결하기전에 특권사용자로 실행하려는 지령을 설정한다.

```
[global] security = value
```

가능한 값은 share, user, server, domain

통과암호보안방법을 설정한다. 만일 security = share로 설정하면 봉사가 공유통과암호로 설정되며 누구나 가능하게 된다.만일 security = user라면 사용자등록자리과 통과암호를 가진다.

만일 security = server로 설정하면 사용자는 등록자리, 통과암호와 함께 Samba에 의한 인증으로 매 콤퓨터를 구별한다. 만일 security = domain으로 설정하면 완전한 NT-domain인증으로 된다.

[global] server string = text

가능한 값은 문자렬 기정값은 Samba %v in 2.0

열람목록에서 봉사기이외에 나타나는 이름을 설정한다.%v(Samba의 판본)와 %h(주콤퓨터이름)변수를 설정한다.

set directory = boolean

가능한 값은 YES, NO 기정값은 NO

[global] smb passwd file = path

가능한 값은 Unix pathname 기정값은 /usr/local/samba/private/smbpasswd 만일encrypted passwords = yes라면 통과암호파일의 경로를 덧쓰기 한다.

[global] smbrun = /absolute_path/command

가능한 값은 smbrun command 기정값은 compiled-in value

smbrun binary 파일의 경로를 덧쓰기 한다. 변경은 피하는것이 좋다.

share modes = boolean

가능한 값은 YES, NO 기정값은 YES

만일 YES로 설정하면 이 추가선택은 Windows-style 로 전체파일을 (deny mode) 만든다.

short preserve case = boolean

가능한 값은 YES, NO 기정값은 NO

만일 YES로 설정하면 의뢰기에 보내는 파일이름을 8.3 style로 만든다.만일 no로 설정하면 기정의 추가선택으로 서술된다.

[global] socket address = IP address

가능한 값은 IP address 기정값은 NULL

련결을 위해서 귀를 귀울이는 IP주소를 설정한다. 기정으로는 모든 주소이다. 하나

의 봉사기에 다중가상대면부를 지원한다.

[global] socket options = socket option list

가능한 값은 list 기정값은 NULL
명백한 소케트추가선택을 설정한다.

[global] status = boolean

가능한 값은 YES, NO 기정값은 YES
만일 YES로 설정하면 일지파일이 smbstatus에 접근가능한 파일에 접속한다.

[global] syslog = number

가능한 값은 number 기정값은 1

[global] syslog only = boolean

가능한 값은 YES, NO 기정값은 NO
만일 yes로 설정하면 표준Samba일지파일이 아니라 syslog만을 기록한다.

[global] time server = boolean

가능한 값은 YES, NO 기정값은 NO
만일 YES로 설정하면 nmbd는 시간봉사를 그 의뢰기에 한다.

unix password sync = boolean

가능한 값은 YES, NO 기정값은 NO
만일 이것을 설정하면 사용자가 자기의 통과암호나 smb통과암호를 변경시키려고 할 때 사용자의Unix password의 변경도 시도하게 된다.

user = comma-separated list

가능한 값은 반점으로 구별된 사용자이름목록.기정값은 NULL

[global] username map = pathname

가능한 값은 경로이름.기정값은 NULL

valid users = user list

가능한 값은 list of users 기정값은 NULL (everyone)
공유시에 기록할수 있는 사용자목록.

veto oplock files = slash-list

가능한 값은 slash-separated list of filenames 기정값은 NULL

```
volume = share name
```

가능한 값은 string 기정값은 NULL
구동기공유 특히 cd-rom의 경우 volume표식을 설정한다.

```
wide links = boolean
```

가능한 값은 YES, NO 기정값은 YES
만일 yes로 설정하면 smaba는 현재 디스크의 기호련결을 마지막까지 실행한다.

```
[global] wins proxy = Boolean
```

가능한 값은 YES, NO 기정값은 NO
만일 yes로 설정하면 nmbd는 낡은 의뢰기의 견지에서 wins봉사기에 대한 요구를 대리해결할것이다. Wins봉사기는 보통 다른 부분망에 있다.

```
[global] wins server = host
```

가능한 값은 hostname 기정값은 NULL
wins봉사기의 dns 이름과 IP주소를 설정한다.

```
[global] wins support = boolean
```

가능한 값은 YES, NO 기정값은 NO
만일 yes로 설정하면 Samba는 wins봉사기를 능동으로 한다. 그러므로 yes로 설정하면 wins server추가선택을 설정하지 말아야 한다.

```
[global] workgroup = name
```

가능한 값은 작업그룹명 기정값은 compiled-in
봉사하려는 작업그룹의 이름을 설정한다. WORKGROUP과 다른 작업그룹이름을 주는것이 좋다.

```
writable = boolean
```

가능한 값은 YES, NO 기정값은 YES
read only의 반대말이며 write ok와 같은 의미를 가진다.

```
write list = comma-separated list
```

가능한 값은 반점으로 구별된 사용자목록. 기정값은 NULL (everyone)
읽기전용공유에 대한 읽기-쓰기접근이 주어진 사용자목록.

```
write ok = boolean
```

가능한 값은 YES, NO 기정값은 YES

```
[global] write raw = boolean
```

가능한 값은 YES, NO 기정값은 YES
64KB의 완충기를 리용하여 tcp상에서 빠른 흐름쓰기를 허락한다.

제3절. Windows로부터 공유등록부를 리용하는 방법

망에 접속되여있는 Windows에서 공유되여있는 등록부를 리용하기 위해서는 특별한 설정이 필요없다. My network의 사용방법을 알고있으며 구동기를 리용하면 된다.
그러면 구체적인 리용방법에 대해 보도록 하자.

6.3.1. Windows의뢰기로부터 공유등록부를 리용

Windows조작체계에서는 봉사기상에 설정되여있는 공유등록부를 리용한다. 이 경우 Windows조작체계의 망설정을 바로 하는것이 중요하다. Windows조작체계를 설치한 콤퓨터에 망이 설정되여있는 경우 desktop상에서는 my network아이콘이 표시된다.

호출권설정이 없는 경우

- desktop에서 my network를 두번 찰칵한다.
- network를 두번 찰칵한다.
- Linux조작체계봉사기의 속성 workgroup itboost를 두번 찰칵한다.
- Linux조작체계를 두번 찰칵한다.
- 공유등록부를 확인한다.

Linux조작체계봉사기가 표시되지 않는 경우 my computer의 오른쪽을 눌러서 콤퓨터의 검색을 선택한다. Linux조작체계의 콤퓨터이름을 입력하여 검색을 진행한다.(my computer상에서 설정이 반영되지 않는 경우도 있다.)

호출권설정이 있는 경우

- 사용자이름이 Windows에 등록가입되였는가를 확인한다.
- 《Start》를 찰칵하면 《Turn Off Computer》 우에 사용자이름의 등록탈퇴가 나타난다. 여기서 사용자이름은 Windows에 등록가입된 사용자이다.
- Windows통과암호입력화면이 표시된다. 여기서 이전에 만들어진 Samba사용자의 이름을 입력하고 OK를 누른다.

공유등록부가 접근가능한가를 확인한다. 사용자가 봉사기에 등록되지 않았다는것을

의미한다. 이때에는 통과암호를 입력하여도 봉사기에 접근할수 없다. Samba에 그 사용자를 등록하여야 한다.

제4절. 관리도구를 리용한 Samba봉사기의 관리

앞의 봉사기들과 마찬가지로 Samba봉사기도 관리도구를 리용하여 구축하고 관리할수 있다. 여기에 대해서는 앞의 부분들을 보면서 어느정도 리해하였으리라고 본다.

봉사기관리도구를 실행시키고 실행된 대면부에서 Samba봉사기부분을 찰칵하면 아래와 같은 대면부가 펼쳐진다.

그림 6-1. 관리도구를 리용한 Samba의 구축

그림 6-1을 보면 알수 있듯이 관리도구를 리용하여 Samba봉사기의 설정에 필요한 모든 작업을 하도록 설계되여있다. 때문에 이 도구를 리용하여 Samba봉사기의 구축을 진행할수 있다.

필요한 설정이 다 되였으면 봉사기를 기동시킨다. 봉사기의 기동과 중지는 아래의 그림과 같은 과정을 거쳐 진행하면 된다.

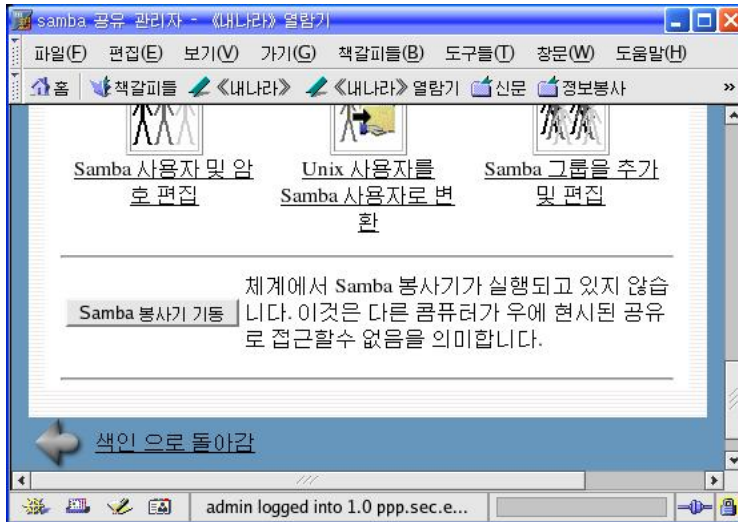



그림 6-2. Samba봉사기의 기동

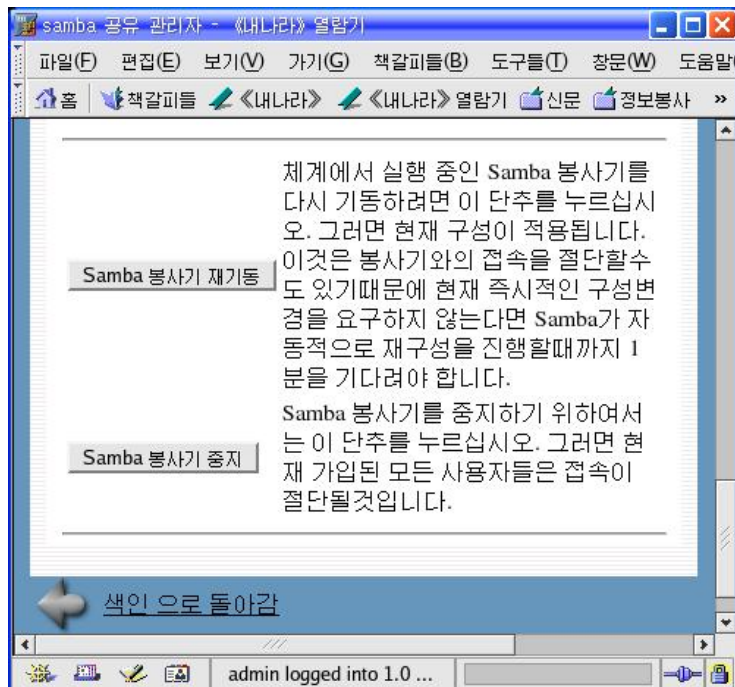



그림 6-3. Samba봉사기의 중지와 재기동

제7장. MySQL봉사기

현시기 자료기지는 프로그람을 작성하는데서 매우 필수적인 요소이다. 우리가 사용하는 자료기지나 자료기지관리라는 용어는 1963년 6월부터 사용되기 시작하였다. 이 시기 자료기지라는 용어는 자료보관장치에 저장된 자료파일을 의미하였다. 그러나 시간이 흐름에 따라 자료기지에 대한 개념은 더욱 확립되여 오늘날 새로운 의미에서의 자료기지개념은 어느 한 집단의 여러가지 조작체계들이 공동으로 사용할수 있도록 통합되고 보관된 자료의 집합이라고 말할수 있게 되였다. 자료기지의 종류는 여러가지 있는데 널리 사용하는 ORACLE, INFORMIX 등이 있는가 하면 개방원천형으로 나온 중소형급 자료기지인 mSQL, PostgreSQL, MySQL 등도 있다.

이 가운데서도 MySQL은 다른 자료기지에 비해 속도가 훨씬 빠르고 사용하기가 편리하며 또한 다중사용자와 다중과제를 지원하고있으므로 거의 모든 Linux사용자들이 리용하고있다 . 이 장에서는 MySQL봉사기의 특징과 거기에서 나오는 여러가지 기능에 대하여 설명한다.

이 장은 아래와 같은 체계로 구성되여있다.

- MySQL봉사기 설치
- MySQL사용법
- MYSQL의 여벌만들기와 MYSQL의 권한체계
- 봉사기관리도구를 리용한 MySQL봉사기의 설치와 관리

제1절. MySQL봉사기 설치

현재 사용되고있는 MySQL의 판본은 mysql-3.23.58이다.
설치고정은 앞에서도 설명한것처럼 2가지 방법이 있다.

7.1.1. RPM에 의한 설치

설치는 간단하게 rpm지령을 리용하여 수행할수 있다.

```
# rpm – ivh MySql*
```

설치가 제대로 되였다면 데몬도 기동된다. 혹시 데몬이 실행되지 않으면 다음의 지령을 실행해보자.

```
# /etc/rc.d/init.d/mysqld start
```

rpm으로 설치하면 실제 자료가 놓이는 등록부는 /var/lib/mysql로 지정되여있다. 자료기지를 창조하면 자료기지이름으로 등록부가 만들어지고 그 아래에 표이름이 만들어진다. 현재는 전체 접근권한에 대한 DB인 mysql이라는 자료기지와 test라는 자료기지가 있다. 그리고 자료기지를 사용하면서 오유에 대한 내용은 hostname.err라는 파일이름으로 하여 오유기록파일에 기록된다. DB에 이상이 발생할 때에는 이 오유기록파일을 확인하면 될것이다.

7.1.2. 원천파일에 의한 설치

원천파일에 의한 설치를 할 때에는 최신판본의 원천파일을 받아서 하는것이 좋다.
MySQL봉사기를 설치하기전에 가장 먼저 해야 할 작업은 MySql데몬을 위한 사용자와 그룹을 추가하는것이다.

```
# groupadd  mysql
# useradd   g  mysql  mysql
```

먼저 원천파일을 가져다 압축을 풀고 설치한다.

```
# tar xvfz mysql*.tar.gz
```

설치등록부로 이동하여 번역(콤파일)환경을 설정하기 위하여 configure추가선택항목에 대해 보도록 하자. 이미 앞의 장들에서 이러한 설치방법에 대하여 보았으므로 구체적인 설명은 생략한다.

```
# mv mysql* mysql-src
# cd mysql-src
# ./configure - - prefix=/usr/local/mysql
```

기본적으로 우와 같은 형태의 추가선택항목들을 많이 사용한다. - - prefix는 설치등록부를 지정한것이다.

우에서와 같이 /usr/local/mysql로 하면 mysql에 대한 모든 내용이 이 등록부밑에 설치된다. 만일 지정하지 않으면 /usr/local로 설치등록부가 지정되여 있기때문에 /usr/local/var, /usr/local/bin등과 같이 /usr/local등록부밑에 나누어 설치된다.

MySQL이 크지 않은 자료기지라고 생각하고 간단하게 사용하지만 용량이 많이 필요하다면 자료기지등록부의 디스크공간을 적당하게 할당하여야 한다. 이때 필요한 추가선택항목이 - -localstatedir추가선택항목이다. 구획들가운데 자료기지를 위한 공간으로만 사용하기 위하여 /data라는 구획이 있다면 다음과 같이 추가선택항목을 설정하여 준다.

```
# ./configure - -prefix=/usr/local/mysql \
-localstatedir=/data \
```

이렇게 한 다음 번역을 진행한다.

```
# make
# make install
```

이렇게 되면 등록부가 생성되고 여기에 파일이 설치되여있는것을 확인할수 있다.

다음으로 자료기지관리자가 사용할 권한표를 만드는것이다. 이것은 스크립트가 있기때문에 실행만 하면 된다.

```
# ./scripts/mysql_install_db
```

그리고 데몬과 모든 자료등록부는 자료기지사용자로 관리하는것이 좋으므로 처음에 생성한 mysql사용자로 자료기지등록부에 대한 권한을 바꾼다. 그리고 기본권한을 root 로 설정한다.

물론 mysql사용자도 권한이 있어야 한다.

```
# chown root    r /usr/local/mysql
# chown mysql    r /usr/local/mysql/var
# chgrp mysql    r /usr/local/mysql
```

우에서 권한을 준 자료기지등록부는 자료기지관리자만이 읽고 쓸수 있는 령역이여야 한다. 만일 일반사용자에게 읽을 권한을 준다면 자료기지의 자료가 쉽게 류출된다.

왜냐하면 mysql은 textbase이기때문에 table file만 편집기로 열어도 쉽게 내용을 확인할수 있다.

설치가 다 되였으면 이제부터 봉사기로 동작하도록 데몬을 실행한다.

간단히 지적하면 자료기지의 이름은 등록부를 의미하는 《/》를 제외한 모든 문자를 허용하며 최대길이는 64자이다. 표의 이름도 최대길이는 64자이다. 그리고 《/》와 《..》를 제외한 모든 문자를 사용할수 있다.

세로표는 모든 문자를 쓸수 있고 64자까지 이름을 지정할수 있다. 표의 별명으로 붙일수 있는 문자는 255자까지 가능하며 모든 문자를 사용할수 있다.

제2절. MySQL사용법

여기에서는 MySQL봉사기를 사용하는 사용법에 대해 설명한다.

일반적으로 자료기지를 사용하는 절차는 MySQL에 접속한후 새로운 자료기지를 창조하고 표를 만든다. 그리고 그 표에 자료를 입력한 후에 필요한 사항에 따라 자료를 추출해낸다. 그러면 이 과정을 차레로 진행하면서 MySql을 습득하도록 하자.

일반적으로 MySQL을 설치하면 자료기지의 root에는 통과암호가 지정되여 있지 않다.

자료기지도 보안상 중요하기때문에 MySql 전체를 관리하는 root 로 설정해야 한다. 만일 이것을 설정하지 않는 경우 모든 사용자가 자료기지의 모든 권한을 가질수 있다. 왜냐하면 MySql 봉사기에 접속할 때에 password항목이 나타나는데 《Enter》건만 누르면 root의 권한으로 자료기지에 접속할수 있기때문이다.

여기서 중요한것은 체계의 root와 자료기지 root의 권한은 엄연히 다르다는것이다.

먼저 MySql봉사기에 접속하여 보자. 사용지령은 다음과 같다.

```
$ mysql -u username -p
Enter password:
```

-u 다음에는 자료기지사용자이름을 지정하고 p 다음에 《Enter》건을 누른후 사용자의 통과암호를 입력한다.

-u 와 p를 지정하지 않고 myslq만을 입력하여 자료기지에 접속하면 현재 사용되는 shell사용자로 접속이 된다. 이 사용자가 권한이 없다면 실제 사용할수 있는것은 일반적인 권한을 준 test라는 자료기지에만 접속할수 있다.

만일 통과암호(password)가 틀리면 아래와 같이 접근할수 없다는 오유통보가 나오면서 접속할수 없다.

```
$ mysql -u root -p
Enter password:
ERROR 1045 : Access denied for user : 'root@localhost'
(Using password:NO)
```

7.2.1. 자료기지만들기(CREATE DATABASE)

MySql을 설치하고 사용하기 위한 기본준비가 끝났으면 이제부터 새로운 자료기지를 만들고 사용하여 보자.

자료기지이름을 member라고 하자.

Show문을 사용하여 현재 봉사기에 있는 자료기지목록을 볼수 있다.

```
Mysql> show databases;
```

Show지령문을 사용한 결과는 아래의 그림과 같다.

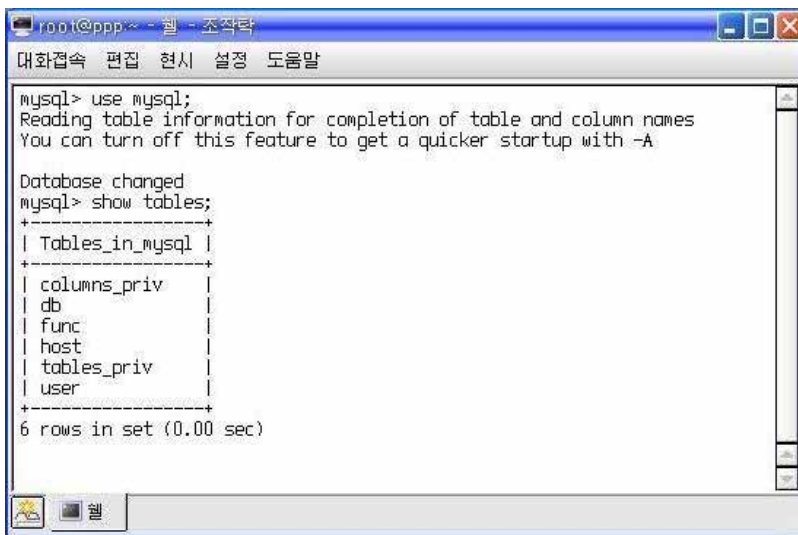



그림 7-1. Show지령문의 사용결과

우에서 보여준 mysql자료기지는 사용자 접근권한정보를 가지고있는 중요한 자료기지이다.

Show 문을 리용하여 그림 7-1과 같이 자료기지의 목록과 표의 목록도 확인할수 있다. 《show tables》를 사용하여 표의 목록을 확인할수 있다.

그러면 이제부터 member라는 자료기지를 만들어 보자.

Unix에서 자료기지이름은 대소문자를 구별한다. 따라서 자료기지이름을 《member》로 하는것과 《Member》, 《MEMBER》로 하는것은 서로 다른 자료기지를 만드는것으로 된다. 마찬가지로 표의 이름도 대소문자를 구별한다.

7.2.2. 표만들기(CREATE TABLE)

자료기지를 만드는것은 매우 쉽다.

자료기지를 다 만들었으면 아래와 같은 지령을 사용하여 표의 목록을 확인해볼수 있다.

```
Mysql> show tables;
Empty set (0.00 sec)
```

《SHOW TABLES;》지령은 선택된 자료기지에 있는 표를 보여주는 지령이다.

table을 만드는것도 자료기지를 만드는것처럼 create문으로 만들면 된다. 실지로 중요한것은 자료기지를 어떻게 설계하는가 하는것이다. 실례로 어느 한 기업소를 대상으로 종업원들에 대한 정보관리를 진행해보자.

종업원의 이름	name
년 령	age
등록번호	serial
직 위	ship
입직년월일	entrance_day
주 소	address
전화번호	tel

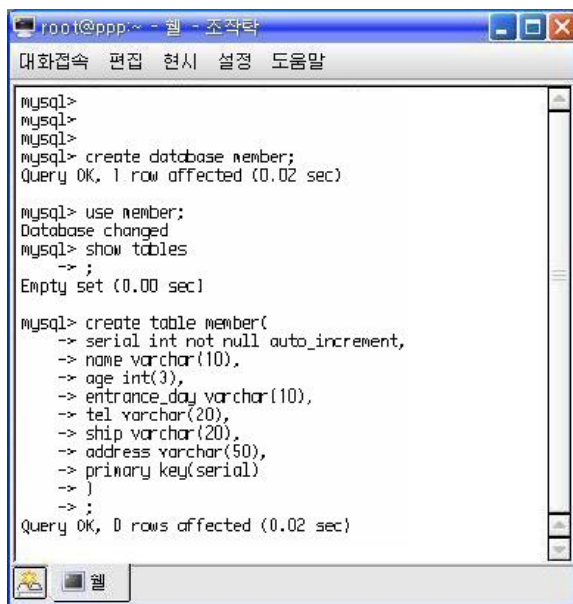



그림 7-2. 자료기지와 표만들기

우와 같이 종업원들의 자료항목이 주어졌다고 하자. 그러면 여기에 기초하여 자료기지의 배렬을 결정하여야 한다.

Create table문으로 표를 만들수 있다. Create table 다음에 표이름을 주고 괄호안에 렬의 이름과 그 렬의 자료형을 한쌍으로 하고 쉼표로 구분하여 렬거하여 준다.(그림 7-2)

문자형은 varchar를 사용하였다.

그리고 primary key 를 넣어 다른 표와 결합할 때 사용한다. 일반적으로 중복이 없는 자료를primary key로 지정한다.

여기에서 mysql만이 가지고 있는 좋은 기능으로는 auto_increment가 있다.

이것은 serial렬에 번호를 자동으로 부여해주고 레코드가 증가할 때마다 번호를 증가시켜준다.

다른 자료기지를 사용한다면 입력할 때 수동으로 증가시켜주어야 하지만 mysql에서는 이 기능을 사용하여 번호를 증가시킬수 있다.

Auto_increment는 한 표에서 한번 밖에 사용할수 없고 not null이여야 한다. 표를 만든 다음 자료기지의 표목록에 추가가 제대로 되였는가를 확인해보아야 한다.

```
root@ppp:~ - 쉘 - 조작탁
대화접속  편집  현시  설정  도움말
mysql> desc member;
+---------------+-------------+------+-----+---------+----------------+
| Field         | Type        | Null | Key | Default | Extra          |
+---------------+-------------+------+-----+---------+----------------+
| serial        | int(11)     |      | PRI | NULL    | auto_increment |
| name          | varchar(10) | YES  |     | NULL    |                |
| age           | int(3)      | YES  |     | NULL    |                |
| entrance_day  | varchar(10) | YES  |     | NULL    |                |
| tel           | varchar(20) | YES  |     | NULL    |                |
| ship          | varchar(20) | YES  |     | NULL    |                |
| address       | varchar(50) | YES  |     | NULL    |                |
+---------------+-------------+------+-----+---------+----------------+
7 rows in set (0.00 sec)

mysql> show columns from member;
+---------------+-------------+------+-----+---------+----------------+
| Field         | Type        | Null | Key | Default | Extra          |
+---------------+-------------+------+-----+---------+----------------+
| serial        | int(11)     |      | PRI | NULL    | auto_increment |
| name          | varchar(10) | YES  |     | NULL    |                |
| age           | int(3)      | YES  |     | NULL    |                |
| entrance_day  | varchar(10) | YES  |     | NULL    |                |
| tel           | varchar(20) | YES  |     | NULL    |                |
| ship          | varchar(20) | YES  |     | NULL    |                |
| address       | varchar(50) | YES  |     | NULL    |                |
+---------------+-------------+------+-----+---------+----------------+
7 rows in set (0.00 sec)
쉘
```

그림 7-3. desc지령을 리용한 표의 확인실례

표가 우에서 지적한대로 만들어졌는지 확인하기 위해서 desc(DESCRIBE)문을 사용한다. Filed 부분과 type부분을 보고 렬의 이름과 자료형을 확인한다.

DESCRIBE는 언제든지 사용할수 있다.

표안의 렬의 이름 및 자료형을 잊어먹었을 경우 사용하면 편리하다.

(Show columns from table도 desc table과 같은 역할을 한다.)
우의 그림 7-3은 desc지령을 리용한 결과를 보여준것이다.

7.2.3. 표에 자료 넣기(INSERT INTO TABLE)

표를 만든 다음에는 표에 자료를 넣어야 한다. 이때 사용하는것이 INSERT지령이다. 좀 더 많은 자료를 한번에 넣자면 MYSQL에서 제공하는 LOAD DATA를 리용할수도 있다.

먼저 INSERT지령을 리용하여 MEMBER표에 자료를 넣어보도록 하자.

한 표에 값을 모두 넣을 때는 항상 렬의 수를 맞게 하여 입력하여야 한다.

사용법은 다음과 같다.

```
Mysql> insert into 표이름 values (값, 값…);
```

이것을 리용한 결과는 그림 7-4와 같다.

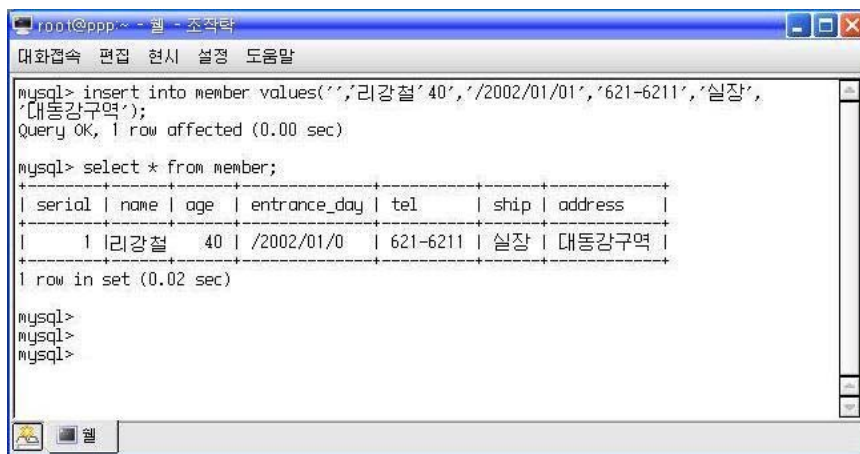



그림 7-4. insert지령의 사용실례

그러면 이제는 우에서 언급된 파일로부터의 자료입력(LOAD DATA INFILE)에 대해 보기로 하자.

그림 7-4에서 보는것과 같이 insert지령을 사용한다면 한번에 한개값씩만 입력이 가능하다. 하지만 종업원정보에 관한것을 Excel파일이나 특정파일로 미리 만들어두었다면 하나하나 입력하는것이 매우 힘들고 품이 많이 드는 일이다. 미리 해놓은것이 없다면 그런대로 참조할수도 있겠지만 파일로 만들어 둔것이 있다면 이것을 리용하는 방법이 더 빠르고 쉬울것은 자명한 리치이다. 한마디로 이전에 이미 작성해둔 파일을 그대로 자료기지에 넣으면 한번에 끝낼수 있기때문이다.

이러한 경우 사용할수 있는것이 load data infile이다. 그의 사용방법은 다음과 같다.

```
Mysql> LOAD D ATA INFILE "파일이름" INTO TABLE 표이름;
```

7.2.4. 표로부터 자료검색하기 (SELECT)

자료의 검색문제는 어디에서나 매우 중요한 문제로 제기되지만 자료기지에서는 특별히 중요한 문제, 사활적인 문제로 제기된다고 볼수 있다. 그것은 자료를 빨리 검색하는것이 자료기지의 리용률을 높이는 가장 중요한 문제로 되기때문이다.

보통 자료의 검색에서는 SELECT 문을 사용한다. 그의 일반적인 사용방법은 아래와 같다.

```
mysql> select 검색대상 FROM 표이름 WHERE 검색조건
```

<검색대상>은 무엇을 보겠는가를 지적하는것이다. 여러개의 렬을 《쉼표》로 구분하여 써넣을수 있으며 모든 렬을 의미하는 《*》를 쓸수도 있다. WHERE부분은 생략할수 있다.

WHERE문을 쓸 때에는 검색조건을 써준다. 검색조건을 만족하지 않는 행은 검색대상에서 제외된다.

- 모든 자료를 검색하기

가장 간단한 SELECT지령으로는 다음과 같은 지령을 사용할수 있다.

```
mysql> SELECT * FROM 표이름;
```

이러한 방법으로 SELECT 문을 사용하는것은 표의 전체정보를 보려고 할 때 유용하다.

이 방법을 리용한 실례를 그림 7-5에 보여주었다.

```
root@ppp:~ - 쉘 - 조작탁
대화접속 편집 현시 설정 도움말
mysql> select * from member;
+--------+--------+------+---------------+-----------+----------+------------+
| serial | name   | age  | entrance_day  | tel       | ship     | address    |
+--------+--------+------+---------------+-----------+----------+------------+
|      1 | 리강철 |   40 | /2002/01/0    | 621-6211  | 실장     | 대동강구역 |
|      2 | 김영남 |   27 | /200/01/01    | 421-6211  | 부원     | 보통강구역 |
|      3 | 김영철 |   28 | /2000/01/0    | 921-6211  | 부원     | 대성구역   |
|      4 | 김철남 |   32 | /1999/01/0    | 453-6211  | 부원     | 평천구역   |
|      5 | 강철수 |   27 | /1999/01/0    | 358-6211  | 부원     | 모란봉구역 |
|      6 | 리은혁 |   27 | /2002/01/0    | 721-6211  | 부원     | 만경대구역 |
|      7 | 송민옥 |   24 | /2004/01/0    | 453-6211  | 책임부원 | 평천구역   |
|      8 | 김은옥 |   19 | /2003/01/0    | 721-6211  | 책임부원 | 만경대구역 |
|      9 | 박향란 |   36 | /1999/01/0    | 621-6211  | 책임부원 | 동대원구역 |
|     10 | 강민철 |   35 | /2002/01/0    | 421-6211  | 부원     | 보통강구역 |
+--------+--------+------+---------------+-----------+----------+------------+
10 rows in set (0.00 sec)
쉘
```

그림 7-5. select문을 리용한 자료의 검색실례(1)

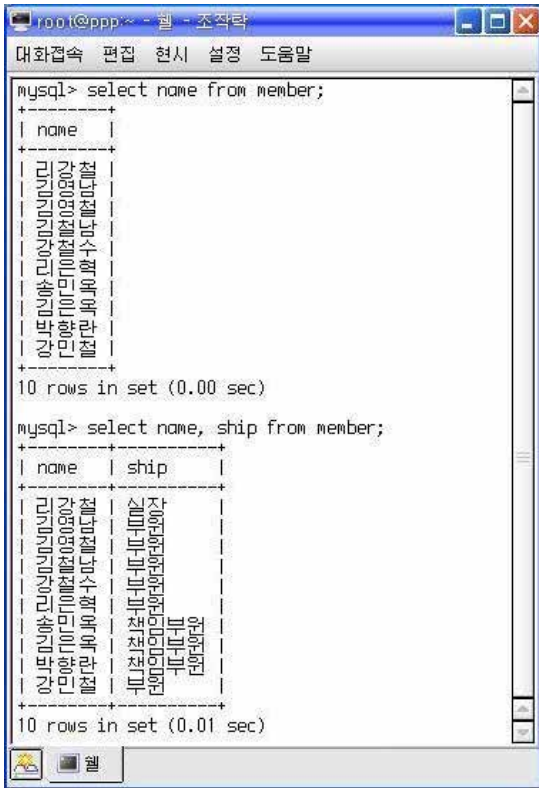



그림 7-6. 조건을 리용한 검색실례

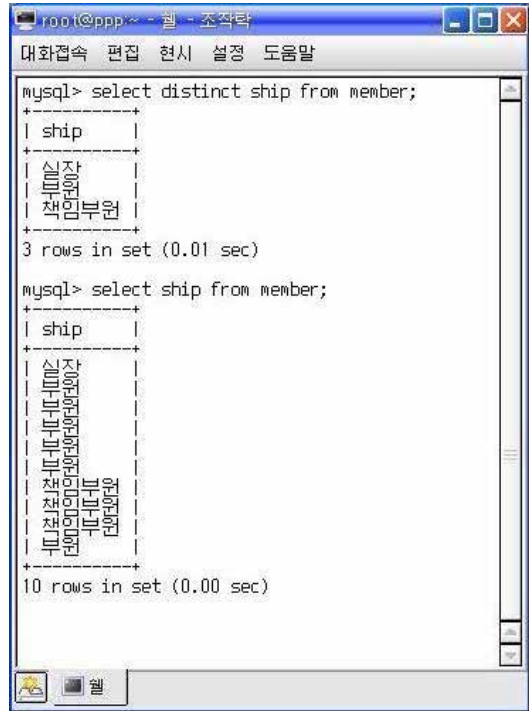



그림 7-7. distinct를 리용한 중복값 제거실례

- 조건을 리용하여 검색하기

아무러한 조건이 없이 검색하는것은 우와 같이 자료가 적을 때에는 별로 문제로 제기되지 않고 보기도 쉽고 필요한 사항을 쉽게 확인할수 있지만 자료의 량이 많아지면 표의 전체내용을 보는것은 시간도 많이 걸리고 원하는 자료를 찾는것이 쉽지 않다. 필요한 사항만 검색하려면 검색조건을 주는것이 좋다.

그러면 주어진 조건에 맞는 특정한 행만을 검색해보자.

표에서 한 행 전체를 보기보다는 필요한 렬만 보고 싶다면 보려는 렬이름을 사용하면 된다. 그림 7-6에 그를 리용한 실례를 들어주었다.

만약 중복된 행이 있을 경우에는 distinct를 사용한다. 그러면 똑같은 값을 뽑은 결과만을 보여준다.

Ship라는 렬에는 직위가 같은 몇개의 자료가 있는데 distinct 를 사용하여 중복값을 뽑은 유일한 값만을 검색하여 준다.(그림 7-7)

그러면 지금까지 배운것을 리용하여 년령이 30이상인 종업원들을 검색하여 보자.

이것은 아래와 같이 지령을 써주면 된다.

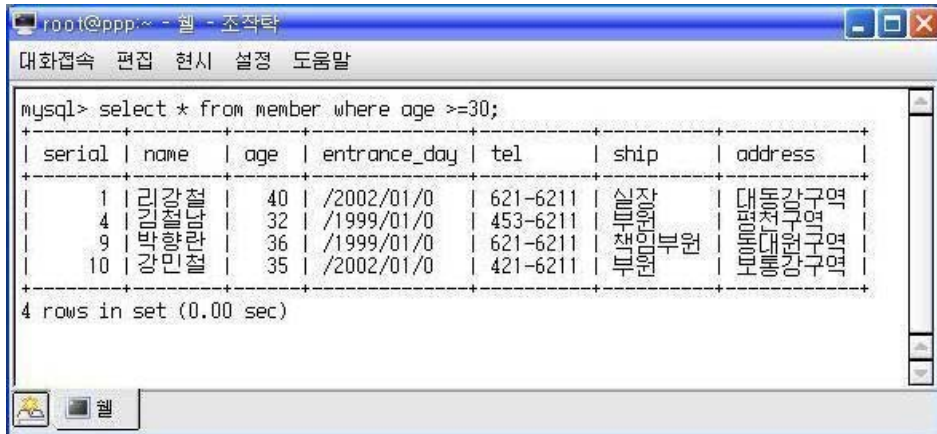



그림 7-8. 년령(age)이 30이상인 자료검색의 실례

만일 다른 조건이 더 주어지면 즉 실례로 년령이 30이상이고 직위가 부원인 종업원을 검색해야 한다면 두가지 조건을 만족해야 하므로 《and》연산자를 사용하면 된다.

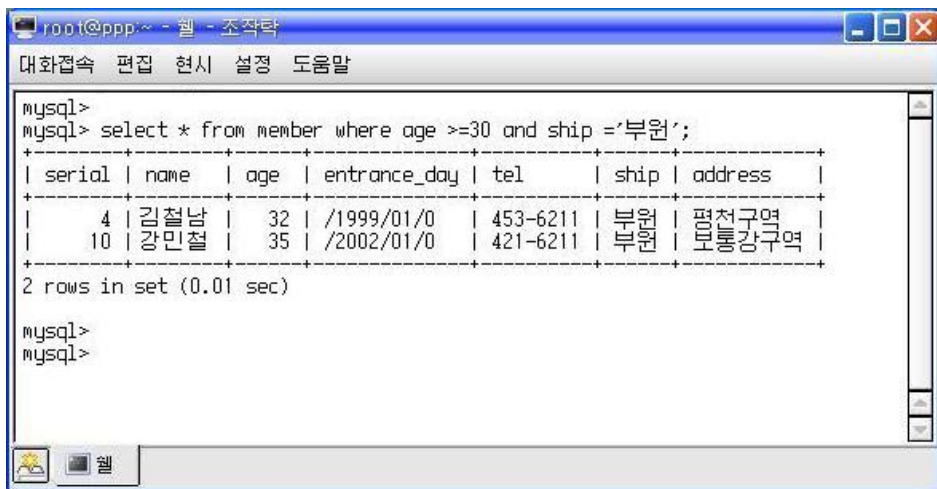
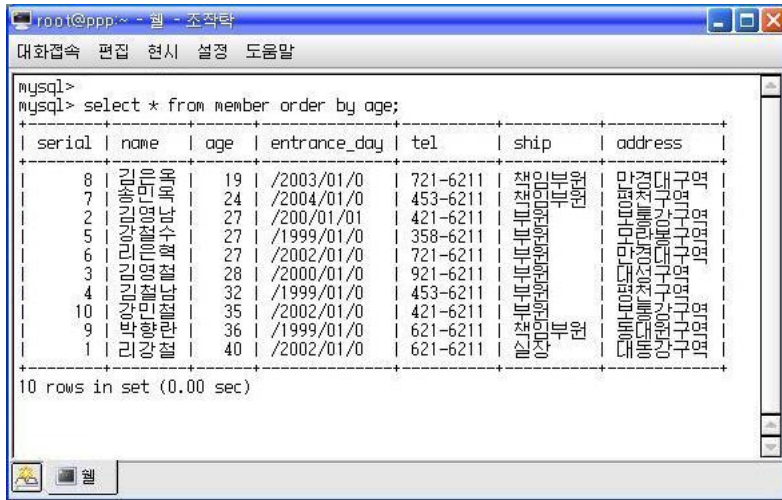



그림 7-9. and연산자를 리용한 복합조건의 검색실례

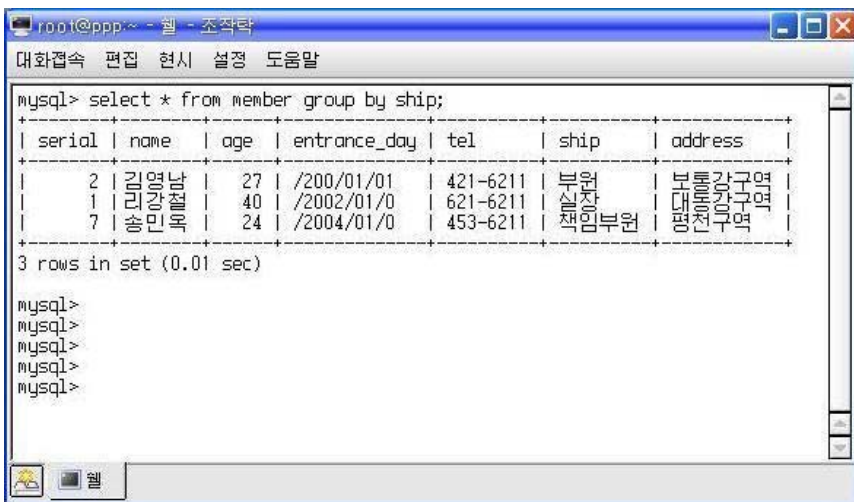
- 행정렬하기

흔히 어떤 자료를 리용하다 보면 자료를 어떤 규칙에 따라 정렬하여야 할 일이 생길 때가 있다. 자료를 일정한 순서로 정렬하려면 ORDER BY를 사용하여 결과를 현시해 본다.

실례를 들어 년령순서로 정렬하여 보자.

```
mysql>
mysql> select * from member order by age;
+--------+--------+-----+--------------+----------+----------+------------+
| serial | name   | age | entrance_day | tel      | ship     | address    |
+--------+--------+-----+--------------+----------+----------+------------+
|      8 | 김은옥 |  19 | /2003/01/0   | 721-6211 | 책임부원 | 만경대구역 |
|      7 | 송민옥 |  24 | /2004/01/0   | 453-6211 | 책임부원 | 평천구역   |
|      2 | 김영남 |  27 | /200/01/01   | 421-6211 | 부원     | 보통강구역 |
|      5 | 강철수 |  27 | /1999/01/0   | 358-6211 | 부원     | 모란봉구역 |
|      6 | 리은혁 |  27 | /2002/01/0   | 721-6211 | 부원     | 만경대구역 |
|      3 | 김영철 |  28 | /2000/01/0   | 921-6211 | 부원     | 대성구역   |
|      4 | 김철남 |  32 | /1999/01/0   | 453-6211 | 부원     | 평천구역   |
|     10 | 강민철 |  35 | /2002/01/0   | 421-6211 | 부원     | 보통강구역 |
|      9 | 박향란 |  36 | /1999/01/0   | 621-6211 | 책임부원 | 동대원구역 |
|      1 | 리강철 |  40 | /2002/01/0   | 621-6211 | 실장     | 대동강구역 |
+--------+--------+-----+--------------+----------+----------+------------+
10 rows in set (0.00 sec)
```

그림 7-10. order by 리용한 검색실례

우의 실례에서는 오름차순으로 검색되였다. 그것은 표준값이 오름차순이기 때문이다.

7.2.5. 그룹화와 함수 사용하기

우와 같은 기능들이외에도 MySQL에서는 여러가지 기능들과 이를 수행할수 있게 하여주는 함수들을 제공하여 준다. 대표적인 기능들과 함수들에 대하여 실례를 들어가며 보도록 하자.

- GROUP BY

다음의 그림 7-11은 렬 SHIP에 대하여 그룹화를 진행한 결과를 실례로 보여준것이다. 그룹화는 자료기지의 사용에서 매우 중요하게 쓰이는 기능의 하나이다.

그룹화는 GROUP BY함수를 리용하여 실현한다.

```
mysql> select * from member group by ship;
+--------+--------+-----+--------------+----------+----------+------------+
| serial | name   | age | entrance_day | tel      | ship     | address    |
+--------+--------+-----+--------------+----------+----------+------------+
|      2 | 김영남 |  27 | /200/01/01   | 421-6211 | 부원     | 보통강구역 |
|      1 | 리강철 |  40 | /2002/01/0   | 621-6211 | 실장     | 대동강구역 |
|      7 | 송민옥 |  24 | /2004/01/0   | 453-6211 | 책임부원 | 평천구역   |
+--------+--------+-----+--------------+----------+----------+------------+
3 rows in set (0.01 sec)

mysql>
mysql>
mysql>
mysql>
mysql>
```

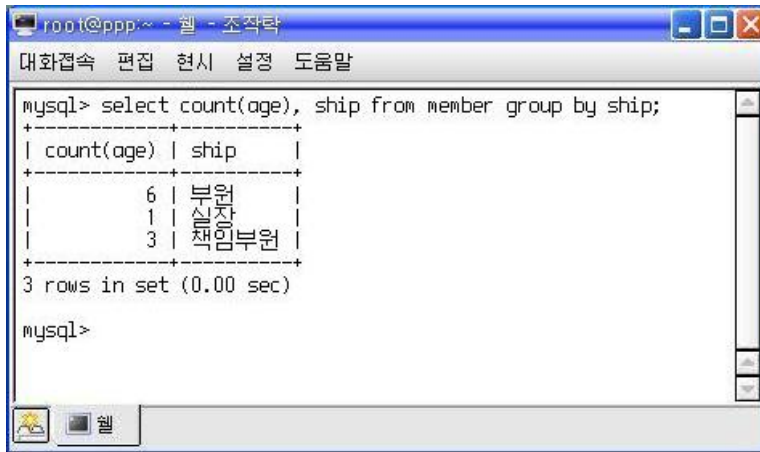



그림 7-11. GROUP BY를 리용한 그룹화의 실례

- COUNT

자료기지를 리용하는 과정에는 매개 렬의 개수를 구하여야 할 때가 생기게 된다. 이러한 경우에는 COUNT함수를 사용할수 있다. 아래의 그림 7-12는 직위별로 그룹화하여 age렬의 개수를 구한것이다.

그림 7-12. count함수의 리용실례

- AVG

어떤 렬의 평균값을 구할 필요가 있을 때에는 AVG(column)를 사용한다.

그림 7-13. avg의 사용실례

- MAX

MAX함수를 사용하면 렬의 최대값을 얻을수 있다.

그림 7-14. Max함수의 사용실례

- MIN

MIN 함수를 사용하면 최소값을 얻을수 있다.

그림 7-15. Min함수의 사용실례

- SUM

SUM은 합계를 구하는 함수이다.

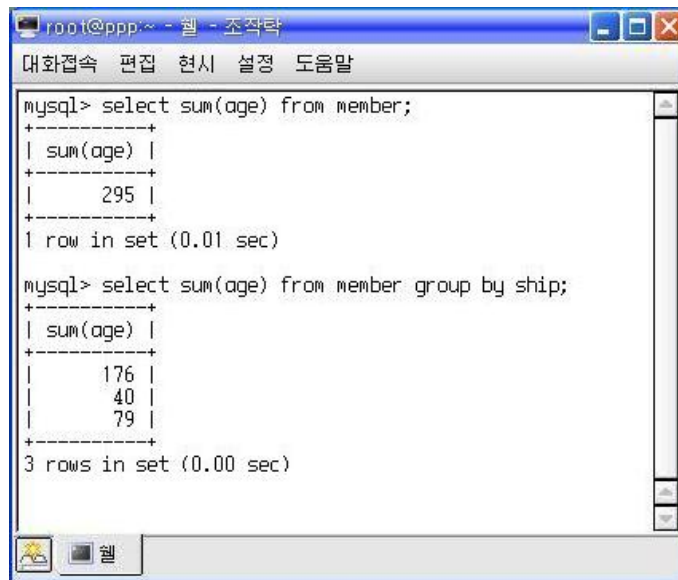



그림 7-16. Sum함수의 사용실레

- 검색결과를 파일에 보관

자료를 입력할 때 LOAD DATA INFILE을 사용한것처럼 검색을 진행한 결과에 대하여 파일로 저장할수 있다.

검색결과를 파일로 보관할 때 사용하는 문법형식은 다음과 같다.

```
Mysql>select * from 표이름 into outfile ‘보관할 파일이름’
Mysql>select * from 표이름 into outfile ‘보관할 파일이름’
fields terminated by ‘구분자’;
```

그러면 그림을 통하여 이 과정에 대해 보자. 그림 7-17은 검색결과에 대한 보관실레이다.

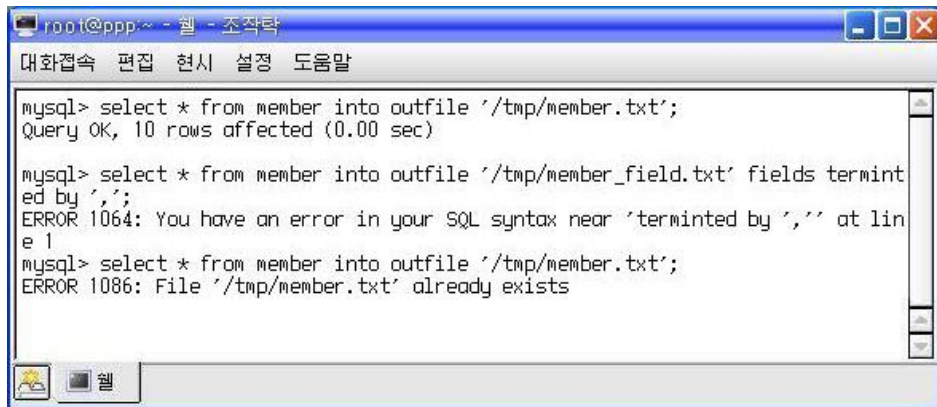



그림 7-17. 검색결과의 파일보관실례

7.2.6. 표안에서 자료 수정하기(UPDATE)

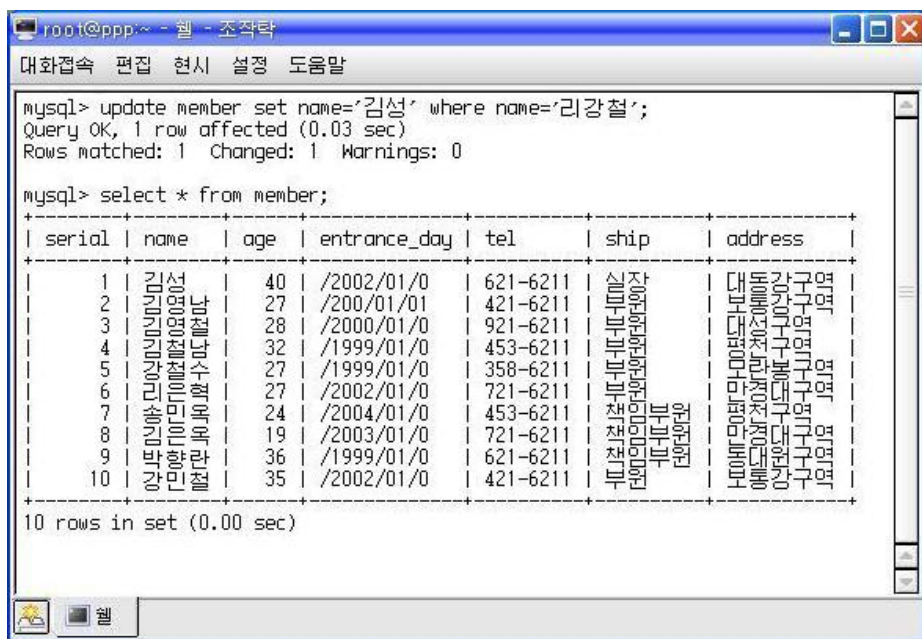



그림 7-18. 표에서 자료의 수정실례

작업을 하다가 보면 자료를 잘못 입력하는 경우가 있다. 그런데 자료를 잘못 입력했다고 해서 자료기지를 다시 작성할수는 없는것이다.

표에 자료를 잘못 입력했을 경우에 수정을 위하여 사용하는것이 UPDATE지령이다. 사용형식은 다음과 같다.

MYSQL> UPDATE 표이름 SET 렬이름=' ' WHERE 조건

그러면 실례를 통하여 수정과정에 대해 보도록 하자.

MEMBER표에서 종업원의 이름이 잘못 되였을 경우 어떤 사람에 대하여 변경해 보자. 자료의 변경시에 항상 주의해야 할것은 조건을 주지 않으면 모든 자료가 변경된다는 것이다. 따라서 특정한 레코드에 대해서만 변경할 경우에는 꼭 조건을 지정해야 한다.

그림 7-18은 MEMBER표에서 NAME이 《리강철》이라고 한것을 《김성》이라고 바꾸는 과정을 실례로 본것이다.

7.2.7. 표안의 자료를 지우기(DELETE)

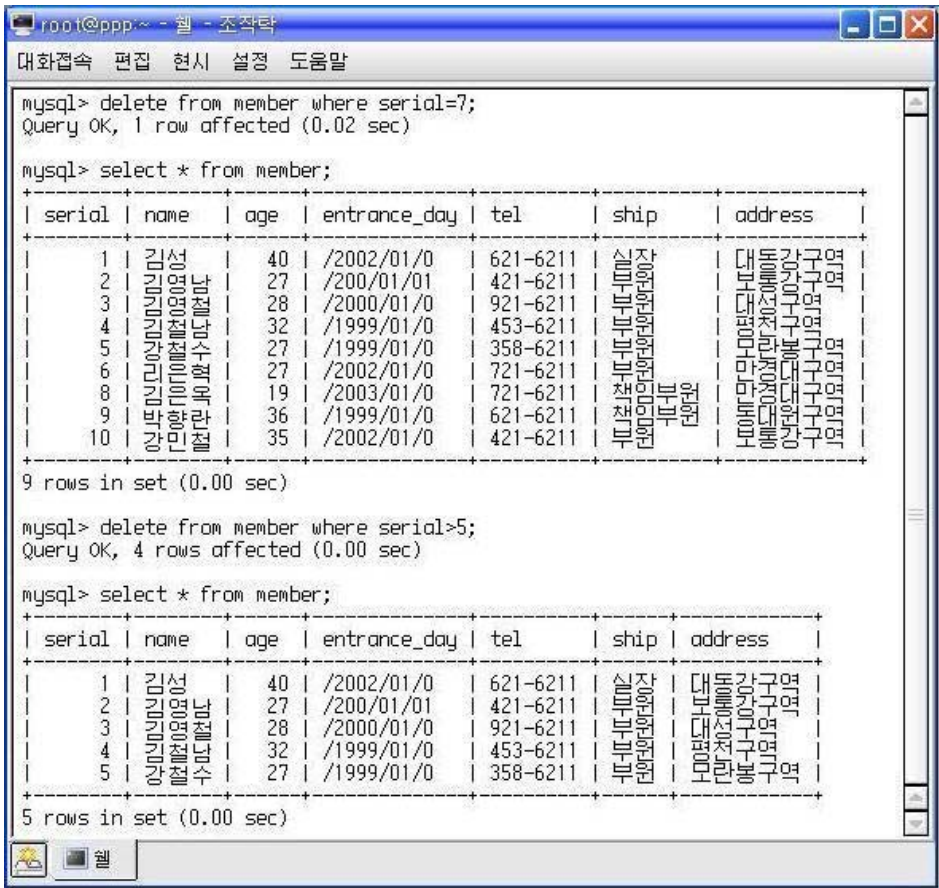



그림 7-19. 표안에서 자료지우기의 실례

특정한 레코드를 지우거나 표의 내용이 모두 잘못되였을때에는 delete지령을 사용하여 잘못된 내용을 삭제한다.

DELETE는 표 자체를 지우는것이 아니라 표의 자료를 지우는것이다. 사용방법은 아래와 같다

```
Mysql> delete from 표이름;
Mysql>delete from 표이름 where 조건;
```

첫번째 지령을 사용하면 표의 내용이 모두 지워지고 두번째 지령을 사용하면 특정한 조건에 맞는것만 지워진다. 간단히 실레를 들어보도록 하자. 그림 7-19는 serial이 7인 것을 지우고 다시 serial이 5보다 큰것을 지우는 과정을 보여준것이다.

7.2.8. 표의 지우기(drop table)

표를 잘못 만들었거나 필요없는 표는 지워도 일없다. 이때 사용하는것이 drop지령이다. 이 지령(DROP나 DELETE)을 사용하는데서 주의하여야 할 점은 실수하는 경우에는 복구할수 없기때문에 어떤 자료라든가 표를 지워버릴 때에는 항상 심중하여야 하며 확인하고 또 확인하여 보아야 한다는것이다.

Drop지령의 사용방법은 다음과 같다.

```
MYSQL>DROP TABLE 표이름;
```

아래의 그림 7-20에서 MEMBER라는 표를 지우는 과정을 실례로 들어 보여주었다.

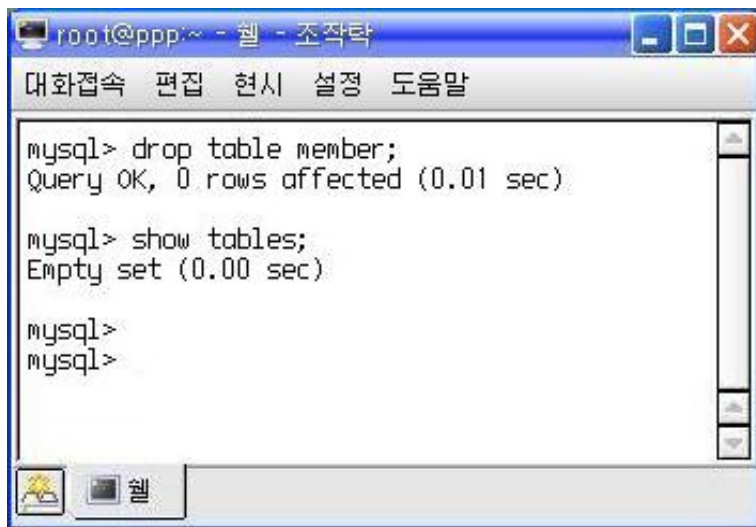



그림 7-20. 표의 지우기

7.2.9. 표의 변경(ALTER TABLE)

```
root@ppp:~ - 쉘 - 조작탁
대화접속 편집 현시 설정 도움말

mysql> desc member;
+--------------+-------------+------+-----+---------+----------------+
| Field        | Type        | Null | Key | Default | Extra          |
+--------------+-------------+------+-----+---------+----------------+
| serial       | int(11)     |      | PRI | NULL    | auto_increment |
| name         | varchar(10) | YES  |     | NULL    |                |
| age          | char(3)     | YES  |     | NULL    |                |
| address      | varchar(30) | YES  |     | NULL    |                |
| entrance_day | varchar(20) | YES  |     | NULL    |                |
| tel          | varchar(20) | YES  |     | NULL    |                |
| ship         | varchar(10) | YES  |     | NULL    |                |
+--------------+-------------+------+-----+---------+----------------+
7 rows in set (0.00 sec)

쉘
```

그림 7-21. 표내용의 확인과정

```
root@ppp:~ - 쉘 - 조작탁
대화접속 편집 현시 설정 도움말

mysql> alter table member add id varchar(10);
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc member;
+--------------+-------------+------+-----+---------+----------------+
| Field        | Type        | Null | Key | Default | Extra          |
+--------------+-------------+------+-----+---------+----------------+
| serial       | int(11)     |      | PRI | NULL    | auto_increment |
| name         | varchar(10) | YES  |     | NULL    |                |
| age          | char(3)     | YES  |     | NULL    |                |
| address      | varchar(30) | YES  |     | NULL    |                |
| entrance_day | varchar(20) | YES  |     | NULL    |                |
| tel          | varchar(20) | YES  |     | NULL    |                |
| ship         | varchar(10) | YES  |     | NULL    |                |
| id           | varchar(10) | YES  |     | NULL    |                |
+--------------+-------------+------+-----+---------+----------------+
8 rows in set (0.00 sec)

쉘
```

그림 7-22. 표에 렬을 추가

한번 생성된 표를 변경하는것은 매우 어려운 일이다. 표의 구조가 복잡할수록 이것은 더 힘든 문제로 제기된다. 그러나 자료기지의 리용과정에는 반드시 안의 내용을 변경하여야 할 때가 꼭 있게 된다. 표에서의 이러한 변경은 alter table지령을 리용하여 진행할수 있다.

```
mysql> alter table member modify id int;
Query OK, 0 rows affected (0.00 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc member;
+--------------+-------------+------+-----+---------+----------------+
| Field        | Type        | Null | Key | Default | Extra          |
+--------------+-------------+------+-----+---------+----------------+
| serial       | int(11)     |      | PRI | NULL    | auto_increment |
| name         | varchar(10) | YES  |     | NULL    |                |
| age          | char(3)     | YES  |     | NULL    |                |
| address      | varchar(30) | YES  |     | NULL    |                |
| entrance_day | varchar(20) | YES  |     | NULL    |                |
| tel          | varchar(20) | YES  |     | NULL    |                |
| ship         | varchar(10) | YES  |     | NULL    |                |
| id           | int(11)     | YES  |     | NULL    |                |
+--------------+-------------+------+-----+---------+----------------+
8 rows in set (0.00 sec)

mysql> alter table member change id id_new varchar(10);
Query OK, 0 rows affected (0.00 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc member;
+--------------+-------------+------+-----+---------+----------------+
| Field        | Type        | Null | Key | Default | Extra          |
+--------------+-------------+------+-----+---------+----------------+
| serial       | int(11)     |      | PRI | NULL    | auto_increment |
| name         | varchar(10) | YES  |     | NULL    |                |
| age          | char(3)     | YES  |     | NULL    |                |
| address      | varchar(30) | YES  |     | NULL    |                |
| entrance_day | varchar(20) | YES  |     | NULL    |                |
| tel          | varchar(20) | YES  |     | NULL    |                |
| ship         | varchar(10) | YES  |     | NULL    |                |
| id_new       | varchar(10) | YES  |     | NULL    |                |
+--------------+-------------+------+-----+---------+----------------+
8 rows in set (0.00 sec)
```

그림 7-23. 렬의 형변환실례

Alter는 현재 있는 렬에 CHANGE, MODIFY, ADD, DROP를 사용하여 추가, 변경, 삭제할수 있게 한다. 그의 사용형식은 다음과 같다.

```
Mysql> ALTER TABLE 표이름 ADD <렬이름> <렬형>;
Mysql> ALTER TABLE 표이름 CHANGE <변경할 렬이름> <새로운 렬이름> <렬형>;
Mysql> ALTER TABLE 표이름 MODIFY <렬이름> <새로운 렬형>;
Mysql> ALTER TABLE 표이름 DROP <렬이름>;
```

그러면 이 지령의 사용방법에 대하여 실레를 들어가면서 보기로 하자.

그러면 그림 7-21과 같이 현재 설치된 member표의 렬을 확인해보자.

표의 내용을 확인한 다음에는 여기에 id라는 렬을 하나 더 추가해보자.

이때 추가한 id렬의 형이 varchar이다. 그러면 이것을 int로 바꾸어보자. 렬을 변경할 때 형만을 변경하려고 한다면 modify를 사용하며 렬이름과 형을 둘다 변경하려고 할 때에는 change를 사용한다.(그림 7-23)

7.2.10. 다른 표와 결합하기(JOIN)

```
root@ppp:~ - 쉘 - 조작탁
대화접속  편집  현시  설정  도움말
mysql> select * from system;
+--------+-----------+------+------+
| serial | cpu       | hdd  | ram  |
+--------+-----------+------+------+
|      1 | 펜티움600 | 10G  | 512M |
|      2 | 펜티움500 | 10G  | 128M |
|      2 | 펜티움600 | 20G  | 1G   |
+--------+-----------+------+------+
3 rows in set (0.00 sec)

mysql> select * from member;
+--------+--------+------+------------+--------------+----------+------+--------
+
| serial | name   | age  | address    | entrance_day | tel      | ship | id_new
|
+--------+--------+------+------------+--------------+----------+------+--------
+
|      1 | 리강철 | 40   | 대동강구역 | /2000/01/01  | 621-6211 | 실장 | 가
|
|      2 | 김영철 | 28   | 대성구역   | /1999/01/10  | 921-2222 | 부원 | 나
|
|      3 | 송민옥 | 24   | 평천구역   | /2004/01/10  | 453-2222 | 부원 | 나
|
+--------+--------+------+------------+--------------+----------+------+--------
+
3 rows in set (0.00 sec)
```

그림 7-24. 두 표의 비교

표를 설계하는 과정에 지켜야 할 중요한 문제의 하나는 련관이 되는 추가선택항목끼리 모아서 될수록 표를 분리하여야 한다. 이런 경우에는 표에 서로 련관된 자료가 존재하게 되는데 이것을 한번에 검색하기 위하여 JOIN을 리용한다.

현재 MEMBER라는 표에는 종업원들의 기본정보가 들어있고 SYSTEM이라는 표에는 매 종업원들이 사용하는 콤퓨터에 대한 정보가 들어있다고 하자. 이 두 표를 련결시켜 주는것이 serial이라는 primary key이다. 현재 두 표에 기록된 정보는 그림 7-24와 같다.

그러면 이제 한 종업원의 기본정보와 체계정보를 모두 한번에 검색해보자.

우와 같이 primary key로 정의된 serial에 따라 종업원들의 정보와 체계정보를 나누어 동시에 입력했다면 두표의 serial값이 같은것을 검색하면 두 표에서 매 종업원들의 정보를 한번에 알수 있다.(그림 7-25)

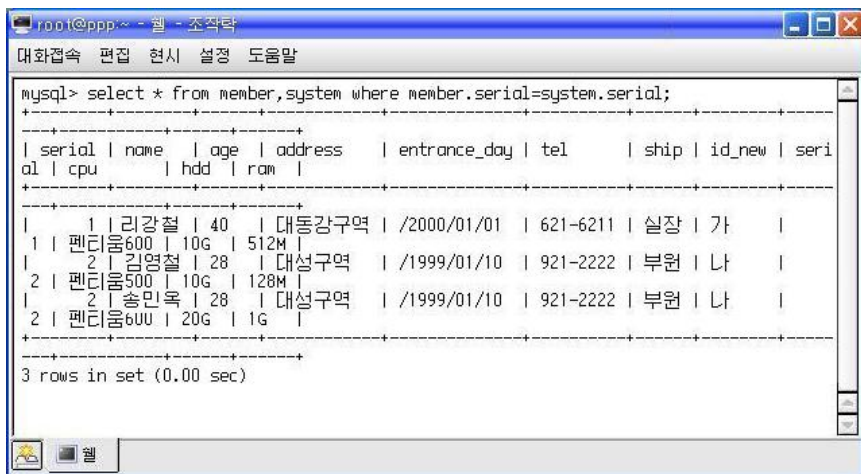



그림 7-25. 두개의 표의 결합실례

이와 같은 기능들에 대해서 잘 알면 자료기지봉사기(여기서는 MySQL)의 구축과 활용을 기본적으로 진행할수 있다. 그러나 이러한 지령들에 대해 잘 모르면 자료기지봉사기를 제대로 활용할수 없게 된다.

그러므로 앞에서 언급된 지령들에 대해 잘 알아두어야 한다.

제3절. MySQL의 여벌만들기와 MySQL의 권한체계

7.3.1. mysql여벌만들기

자료기지를 관리해보면 매우 중요한것이 자료의 여벌만들기이다.

여벌만들기의 중요성에 대해서는 콤퓨터를 어느정도 해본 사람이라면 누구나 다 체험을 통하여 알고있으리라고 본다. 힘들게 구축하여 놓은 체계가 《비루스》나 기타 다른 원인으로 파괴되였을 때 그안의 자료가 모두 류실되는 실례는 아주 많다. 이 경우 자료를 여벌로 만들어 놓지 않았다면 모두 없어진다. 그러므로 자료의 여벌복사는 매우 중요한 문제로 제기된다.

그러면 지금까지 련습과정으로 구축해 놓은 자료기지를 여벌복사해 보도록 하자. 이때 사용되는 지령이 mysqldump지령이다.

이 지령은 아래와 같은 형식으로 사용된다.

```
$mysqldump  u root  p member > member.sql
Enter password
```

우와 같이 지령을 주어 실행을 시키면 member자료기지에 있는 모든 표와 자료가 member.sql파일에 보관된다.

7.3.2. mysql권한 system

지금까지 MYSQL 을 사용하면서 사용자이름과 통과암호를 사용한 인증을 통하여 접속을 진행하고 자료기지를 만들고 표를 만들어 자료를 입력했다. 그리고 자료기지를 창조할 때에 아무러한 문제도 없이 표와 자료를 만들고 쓸수 있는 권한을 리용하였는데 이것은 root로 접속했기때문이다.

한마디로 mysql을 사용하는데서 사용자에 따라 어떠한 권한이 주어진다는것을 알수 있다. 그러면 이에 대한 모든 설정이 어디에 있으며 그리고 접속할 때의 사용자라는 개념이 무엇인지를 알기 위하여 mysql의 권한체계에 대해 보도록 하자.

Mysql권한체계의 주요기능은 자료기지에서 select,insert,update, delete권한을 콤퓨터사용자의 이름과 련관시키는것이다. 권한과 관련된 정보는 mysql자료기지의 user,db,host,Table_priv,colums_priv표에 보관된다.

앞에서 Mysql을 처음 설치한 다음 mysql등록부를 확인하였다. 처음 설치하면 기본사용자인 root 와 일반적인 정보가 모두 이 자료기지안에 기록된다. 먼저 권한정보가 들어있는 mysql자료기지의 표를 확인해 보도록 한다.(그림 7-26)

만일 권한설정을 잘못하여 일반사용자가 접근하게 되면 통과암호나 자료의 손실을 가져 올수 있다. 이것은 봉사기관리자나 사용자에게 있어서 매우 무서운 일이며 반드시 허용해서는 안될 문제이다. 현재 기본적인 권한인 user와 password정보가 들어있는 표는 user표이다.(그림 7-27)

그림 7-26. 권한설정표의 확인

그림 7-27. user표의 내용

그러면 매개 렬에 대해 보도록 하자.

Select, insert, update, delete권한은 자료기지표에서 레코드에 대한 조작을 할수 있도록 허용한다. Select문은 실제로 표에서 레코드를 가져올 때에만 select 권한이 필요하다. 봉사기의 자료기지에 접근할수 있는 권한이 없는 경우에도 특정한 select문은 사용할수 있다.

Index권한은 index를 창조하거나 제거할수 있게 하여준다. Alter권한은 alter table을 사용할수 있도록 한다. Create 와 drop권한은 새로운 자료기지와 표를 창조하거나 존재하는 자료기지와 표를 제거할수 있도록 허용한다. 사용자에게 mysql자료기지의 drop권한을 허용하면 그 사용자는 mysql접근권한 정보가 보관된 자료기지를 삭제할수 있다. Grant권한은 사용자가 가지고 있는 권한을 다른 사용자가 가질수 있도록 허용한다. File권한은 load data infile and select… into outfile문을 리용하여 봉사기에 파일을 보관하고 읽을수 있는 권한을 허용한다. 나머지 권한들은 관리자의 권한을 위하여 사용되며 mysql admin프로그람을 사용하여 실행한다. Mysql admin으로 권한을 적용시킬수 있는 지령어는 다음과 같다.

표 7-1. 권한과 지령과의 관계

권 한	지 령 어
reload	reload, refresh, flush-privileges, flush-hosts, flush-logs, flush-tables
shutdown	Shutdown
process	processlist, kill

Reload지령은 봉사기가 grant표를 다시 읽어들이도록 하는 지령이다. Refresh지령은 열려진 모든 표를 닫으며 일지기록파일을 열고 닫는다. Flush-privileges는 reload지령과 같은 역할을 하는것이다. 다른 flush-*지령들은 refresh와 비슷한 기능을 수행한다. Shutdown지령은 mysql봉사기를 완료한다.

Processlist지령은 봉사기에서 실행되고 있는 모든 처리에 대한 정보를 나타낸다. Kill지령은 봉사기의 프로쎄스를 죽인다. Processlist지령도 될수록이면 root사용자만이 사용할수 있도록 하는것이 좋다.

다음은 권한설정시에 주의해야 할 사항이다.

- grant권한은 사용자가 다른 사용자의 권한을 설정할수 있도록 허용한다. 다른 권한과 grant권한을 가진 두 사용자는 권한을 결합할수 있다.
- alter권한을 잘못 사용하면 권한표의 표이름을 바꾸어서 권한체계자체를 파괴할수도 있다.
- file권한은 자료기지표안에서 select문을 리용하여 접근할수 있기때문에 봉사기의 파일을 모든 사람들이 읽을수 있는 파일로 되게 할수 있다.
- shutdown권한을 사용하면 다른 사용자가 완전히 봉사를 받지 못하도록 할수 있다.
- process권한은 비밀번호를 설정하고 바꾸는 질문을 포함하여 현재 수행하고있는 질문을 보는데 사용할수 있다.
- mysql자료기지에 대한 권한은 비밀번호와 다른 접근권한정보를 바꾸는데 사용할수 있다.

Mysql권한 체계로는 다음과 같은것들은 할수 없다.

- 접근을 거부할 사용자를 명백하게 지정할수 없다. 왜냐하면 사용자와의 련결을 거부하는것을 완전하게 정합시킬수 없기 때문이다.
- 사용자가 자료기지에서 표를 만들고 지울수 있는 권한을 가질수 있지만 자료기지자체를 만들고 삭제할수는 없도록 지정할수 없다.

7.3.3. 권한체계의 사용

MySQL은 사용자를 확인하는데서 콤퓨터이름과 사용자이름 모두를 확인한다. Mysql자료기지의 user표에서도 host, user, password항목을 확인하였을것이다.

그러면 MySQL에서의 권한체계에 대해 보도록 하자.

MySQL의 접근조종에는 두가지 단계가 있다.

- 단계1:봉사기에서 사용자가 련결할수 있는지, 없는지 판단.
- 단계2:봉사기에 사용자가 련결이 허용되였을 경우 사용자가 수행하려는 지령에 대해 충분한 권한이 있는지 봉사기에서 판단.

련결요청을 하는 경우 user표에서 사용자와 통과암호에 대해 인증을 제대로 하였어도 즉 정상적으로 련결되였어도 select를 하려면 사용자에 대하여 표에 접근할수 있는 권한이 있는가를 검사해야 하며 표를 만들려고 하면 권한표에 create_priv가 있는가를 검사하여야 한다.

봉사기는 접근조종의 두가지단계에서 mysql자료기지의 user, db, host표를 리용하여 접근조종을 관리한다. 접근조종의 두번째 단계를 위해 제기되는 요청이 표에 관계된것이라면 추가적으로 tables_priv와 columns_priv표를 참조한다.

권한표에 대해 간단히 설명하면 다음과 같다.

1. user표의 범위마당인 host, user, password는 들어오는 련결에 대해 허용할것인지 거부할것인지를 결정한다. 허용된 련결에 대하여 권한마당이 select_priv, insert_pric, update_priv, delete_priv, create_priv, reload_priv, shutdown_priv, process_priv, file_priv, grant_priv, references_priv, index_priv, alter_priv는 사용자의 전체적인 권한을 가리킨다.

2. db와 host표는 함께 사용된다.

자료기지표의 host,db, user는 어떤 host에서 어떤 자료기지에 대해 어떤 사용자가 접근할수 있는지를 결정한다. 권한마당은 어떤 조작이 허용되였는가를 결정한다. Host표는 DB표의 entry를 여러개의 host에 적용하려고 할 때 DB표의 확장을 위하여 사용한다. 례를 들어 사용자가 현재 망우에서 여러 host의 자료기지를 사용할수 있도록 하려면 사용자의 db table_entry에 host값을 비워두고 host표에 각 host의 entry를 넣으면 된다.

3. tables_priv와 colums_priv표는 db표와 비슷하다. 그렇지만 더 세부적으로 지정할수 있다.이 표들은 자료기지단계에서 더 나아가서 표와 렬단계에 적용할수 있다.

Admin 권한인 db reload,shutdown은 오직 user표에서만 지정할수 있다. 파일권한도 마찬가지로 user 표에서만 지정한다.

접근조종, 단계 1: 련결확인

MySQL봉사기에 접속하려고 할 때 사용자확인과 비밀번호를 통해 접속을 허용하거나 거부한다. 사용자의 확인이 되지 않으면 봉사기는 접속을 완전히 거부한다.

사용자가 확인되면 봉사기는 련결을 받아들이고 봉사기는 접속을 완전히 거부한다. 사용자가 확인되면 봉사기는 련결을 받아들이고 2번째 단계로 들어가며 요청을 기다린다.

- 접속하는 주콤퓨터
- MySQL사용자이름

사용자확인은 user표의 세가지 범위마당(Host, User, Password)을 사용하여 수행된다. 봉사기는 user table_entry의 host이름과 사용자이름이 맞고 비밀번호가 정확할 때에만 접속을 받아들인다. 아래와 같이 user표의 범위마당값을 지정할수 있다.

1. Host값은 주콤퓨터 이름이나 IP 또는 local host일 경우는 'localhost'가 될수 있다.
2. Host마당에서 '%'와 '_'의 문자를 사용할수 있다.
3. Host설정시 MySQL 3.23판본부터는 IP대신 많은 주콤퓨터를 허용할 경우 망범위로 지정할수 있다. 례를 들면 192.168.1.0/255.255.255.0과 같은 형식으로 지정할수 있다.
4. '%'의 Host값은 모든 주콤퓨터 이름을 나타낸다.공백의 주콤퓨터 값은 '%'와 같다.
5. 와일드 카드문자는 User마당에는 허용되지 않는다.하지만 모든 사용자를 설정할 경우에는 공백으로 할수 있다.
6. Password마당은 공백이 될수 있다. 하지만 이것은 아무 비밀번호나 모두 사용할수 있는것이 아니라 처음에 설치하고 enter를 입력했던것처럼 지정하지 않고 입력을 해야 한다는것을 의미한다.

접근조종, 2단계: 요청확인

련결되였다면 봉사기는 두번째단계로 들어간다. 련결이 이루어졌을 때 개개의 요구에 대해 사용자가 수행하려는 연산에 따라 봉사기는 사용자가 충분한 권한을 가지고 있는지 검사한다.

여기서 승인표의 권한마당이 동작한다. 권한은 user, db, host, table_priv, columns_priv표의 정보를 사용한다.

GRANT와 REVOKE지령을 리용하여 권한표를 다룰수 있다.

USER TABLE의 승인권한은 사용자에게 전체적인 권한을 제공하며 현재의 자료가 어떤것인가는 상관없다. 례를 들어 USER표에서 사용자에게 delete 권한을 승인했다면 전체 mysql의 어떤 자료기지의 record라도 삭제할수 있다. 다르게 말하여 user table권

한은 super user권한이며 봉사기나 자료기지관리자에게만 user표에 대한 권한을 승인하는것이 좋다. 다른 사용자에게는 User table에서 권한을 N으로 설정하고 db와 host표를 사용하여 특정자료기지에 기초한 권한승인을 하는것이 보안상 필요하다.

Db와 host 표는 특정한 자료기지의 권한을 허가한다. 매 표의 마당에 대해 다음과 같이 설명할수 있다.

1. host와 db마당에 문자 '%', '_'를 사용할수 있다.
2. db 표에서 host값의 '%'는 모든 host를 의미한다. 공백은 모든 값을 허용한다.
3. host 표의 공백이나 '%'은 모든 host를 의미한다.
4. host table에서 db의 값이 '%'나 공백이면 모든 자료기지를 의미한다.
5. user값이 공백이면 닉명사용자로 간주한다.

봉사기가 시작할 때 db와 host표를 읽고 정렬한다. 동시에 user table을 읽고 정렬한다. DB table은 host, db, user순서로 마당범위를 정렬하며 host표는 host, db순서로 마당범위를 정렬한다. User table과 같이 특정하게 지정되여 있는값이 먼저 정렬되고 전체 범위로 지정된 값이 나중에 정렬된다.

봉사기에서 인증은 user table과 마찬가지로 가장 먼저 정합되는것을 사용하여 진행한다.

Table_priv와 columns_priv표는 특정한 표와 렬에 관련된 권한을 승인한다.

권한변경시 적용

GRANT,REVOKE,SET PASSWORD를 리용하여 모든 승인표에서 변경을 하면 봉사기에서 인식한다. 권한표를 직접 변경했다면 (INSERT,UPDATE 등을 사용하여) 봉사기에서 승인표를 재시동하도록 하기 위해 FLUSH PRIVIEGES문이나 mysqladmin flush-privileges, Mysqladmin reload를 실행해야 한다.

그렇게 하지 않으면 봉사기를 다시 시작하기전까지는 변경된 권한이 적용되지 않는다. Root의 비밀번호를 바꾼것을 확인해보자.

```
$ mysql -u root myslq
mysql>UPDATE user SET password=PASSWORD('new_password') WHERE user='root';
mysql>FLUSH PRIVILEGES;
```

봉사기에서 권한표가 변경되였다는것을 알았을 때 이미 존재하던 client의 련결은 다음과 같은 영향을 받는다.

1. table과 column권한 변경은 client의 다음 요청부터 적용된다.
2. 자료기지 권한 변경은 다음의 USE db_name지령부터 적용된다.
3. 전체적인 권한과 비밀번호 변경은 client가 다음에 련결할 때부터 적용된다.

7.3.4. MySQL에 새로운 사용자 권한 추가하기

두가지 방법으로 사용자를 추가할수 있다. GRANT문을 사용하거나 mysql권한표를 직접 추가하거나 변경하는것이다. 여기에서는GRANT문을 사용하여 사용자를 추가해보도록 하자. 먼저 root사용자로 mysql에 접속한다.

아래의 그림은 grant로 coffeein과 admin사용자를 생성하고 권한을 설정한것이다. 그리고 권한을 적용하기 위해 flush privileges를 사용했다. 권한표를 직접 수정할수 있기때문에 GRANT를 리용하여 설정한것이 어떻게 적용되였는지 확인하기 위해 USER표를 확인한다.

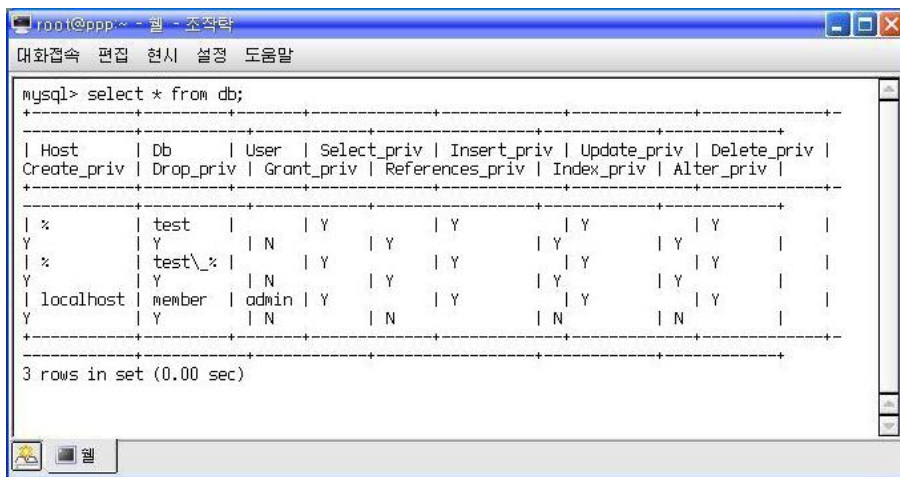
```
root@ppp:~ - 쉘 - 조작탁
대화접속  편집  현시  설정  도움말
mysql> grant all privileges on *.* to coffeein@localhost identified by 'coffein' with grant option;
Query OK, 0 rows affected (0.00 sec)

mysql> grant usage on *.* to admin@localhost identified by 'admin';
Query OK, 0 rows affected (0.00 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)

mysql> use mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from user
    -> ;
+-----------+----------+------------------+-------------+-------------+-------------+-------------+
-------------+-----------+-------------+---------------+--------------+-----------+------------+---
-------------+------------+------------+
| Host      | User     | Password         | Select_priv | Insert_priv | Update_priv | Delete_priv |
 Create_priv | Drop_priv | Reload_priv | Shutdown_priv | Process_priv | File_priv | Grant_priv | Re
ferences_priv | Index_priv | Alter_priv |
+-----------+----------+------------------+-------------+-------------+-------------+-------------+
-------------+-----------+-------------+---------------+--------------+-----------+------------+---
-------------+------------+------------+
| localhost | root     |                  | Y           | Y           | Y           | Y           |
 Y           | Y         | Y           | Y             | Y            | Y         | Y          | Y
              | Y          | Y          |
| won       | root     |                  | Y           | Y           | Y           | Y           |
 Y           | Y         | Y           | Y             | Y            | Y         | Y          | Y
              | Y          | Y          |
| localhost |          |                  | N           | N           | N           | N           |
 N           | N         | N           | N             | N            | N         | N          | N
              | N          | N          |
| won       |          |                  | N           | N           | N           | N           |
 N           | N         | N           | N             | N            | N         | N          | N
              | N          | N          |
| localhost | coffeein | 7bd58a846da664fb | Y           | Y           | Y           | Y           |
 Y           | Y         | Y           | Y             | Y            | Y         | Y          | Y
              | Y          | Y          |
| localhost | admin    | 43e9a4ab75570f5b | N           | N           | N           | N           |
 N           | N         | N           | N             | N            | N         | N          | N
              | N          | N          |
+-----------+----------+------------------+-------------+-------------+-------------+-------------+
-------------+-----------+-------------+---------------+--------------+-----------+------------+---
-------------+------------+------------+
6 rows in set (0.00 sec)
```

그림 7-28. GRANT의 실례

Coffeein사용자는 root 와 같이 localhost에서 'coffeein'이라는 비밀통과암호를 사용하여 접속해야 하고 모든 권한표에 대한 권한을 주었다. 그리고 admin사용자를 일반사용자로 하기 위해 localhost에서 접속할수 있고 비밀통과암호는 'admin'을 사용하여 접속할수 있다.

또한 USAGE를 주었기때문에 아무런 권한도 없다.

때문에 이제는 admin사용자가 자료기지를 사용할수 있도록 db표에서 사용자권한을 줄수 있다.

그러면 이제 admin사용자에게 자료기지를 사용할수 있도록 권한을 부여해보자.

권한의 부여는 아래와 같은 과정을 통하여 진행할수 있다.

```
mysql>grant    select,insert,delete,update,create,drop    on
member.* to admin@localhost  Identified by  'admin' ;
  Query OK, 0 rows affected (0.01 sec)
```

우의 과정은 Member자료기지의 localhost에서 admin이라는 사용자로 admin통과암호를 사용하여 select, insert, delete, update, create, drop할수 있는 권한을 지정하였다.

권한표의 변경내용을 확인하여보자.

User표에 이미 admin이 추가되였기때문에 db표를 확인하면 된다.(그림 7-29)

mysql> select * from db;

Host	Db	User	Select_priv	Insert_priv	Update_priv	Delete_priv	Create_priv	Drop_priv	Grant_priv	References_priv	Index_priv	Alter_priv
%	test		Y	Y	Y	Y	Y	Y	N	Y	Y	Y
%	test_%		Y	Y	Y	Y	Y	Y	N	Y	Y	Y
localhost	member	admin	Y	Y	Y	Y	Y	Y	N	N	N	N

3 rows in set (0.00 sec)

그림 7-29. db표의 권한변경확인

제4절. 봉사기관리도구를 리용한 MySQL봉사기의 설치와 관리

앞의 절들에서도 설명한것처럼 봉사기관리도구를 리용하여 봉사기들을 설치하고 관리할수 있다. 봉사기의 설치과정은 이미 앞에서 설명하였으므로 다시 설명하지 않는다. 봉사기관리도구의 대면부에서 MySQL봉사기에 해당한 부분을 선택하면 아래와 같은 대면부가 나타난다.

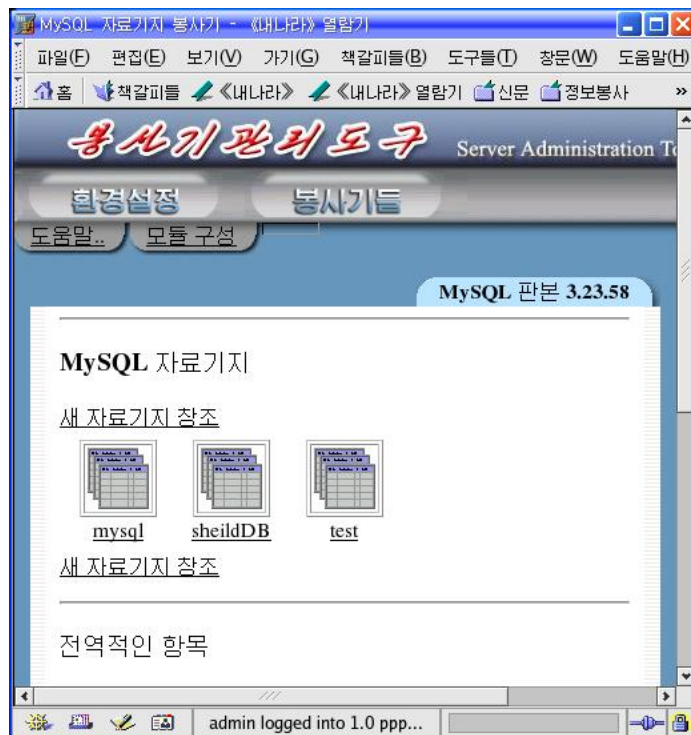



그림 7-30. 봉사기관리도구를 리용한 봉사기관리(1)

우의 그림에서와 같은 대면부가 나타나면 해당한 항목들을 선택하여 봉사기를 구축하고 필요한 항목들을 설정할수 있다.

구체적인 부분들은 아래의 그림들과 같다.

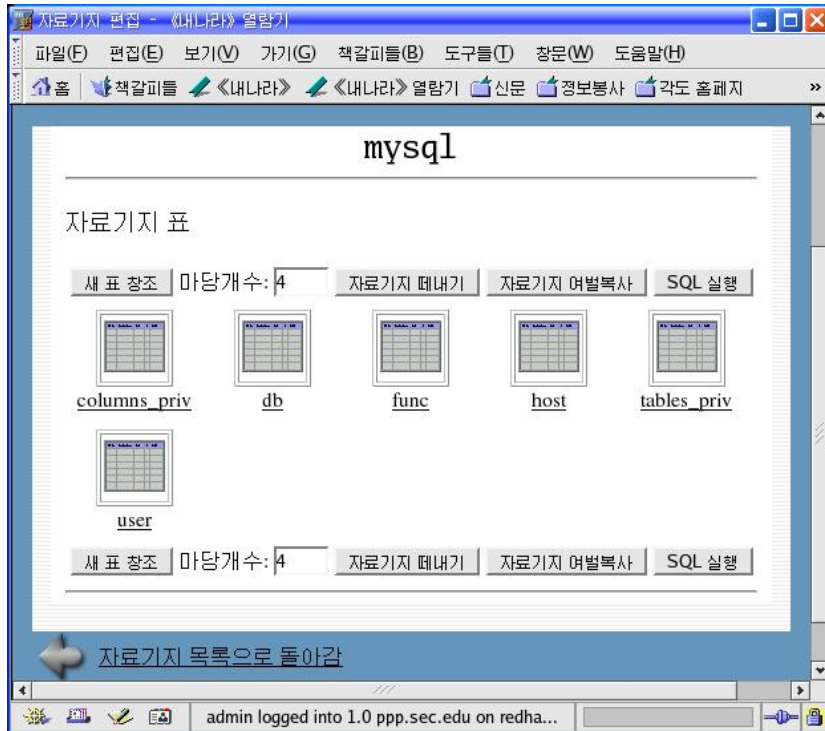



그림 7-31. 봉사기관리도구를 리용한 봉사기관리(2)

그림 7-32. 봉사기관리도구를 리용한 봉사기관리(3)

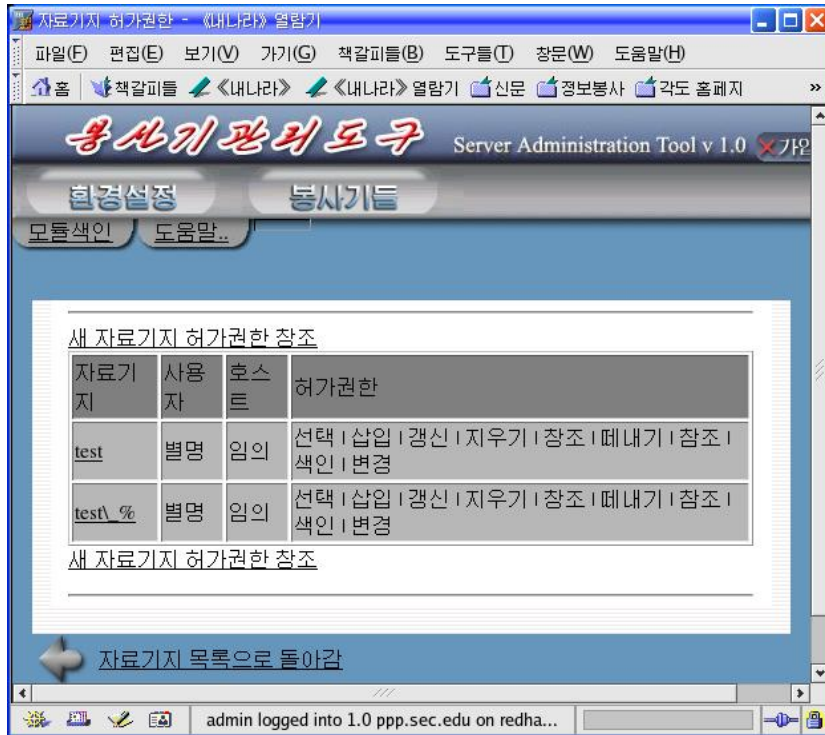



그림 7-33. 봉사기관리도구를 리용한 봉사기관리(4)

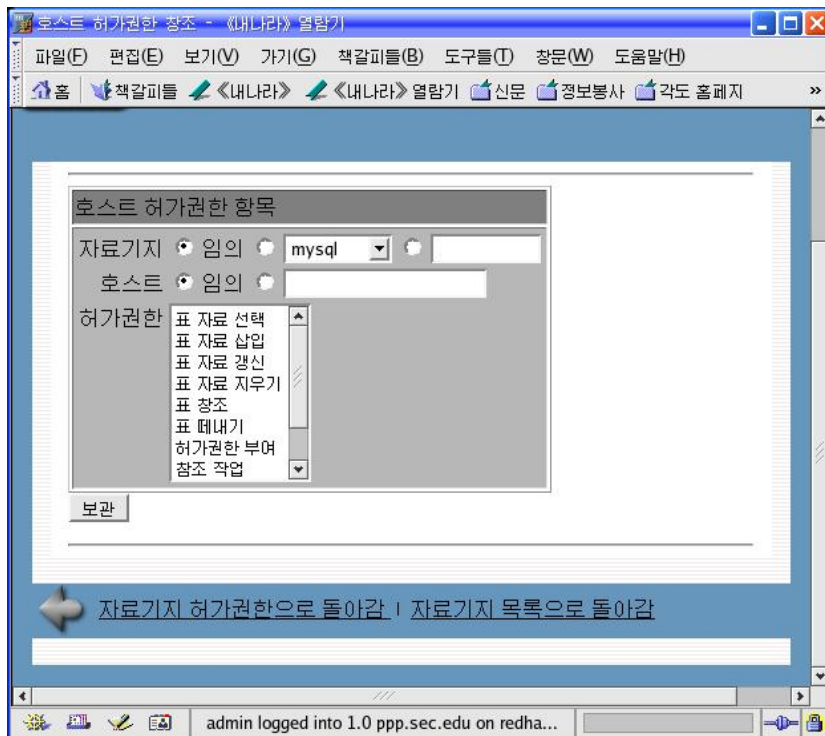



그림 7-34. 봉사기관리도구를 리용한 봉사기관리(5)

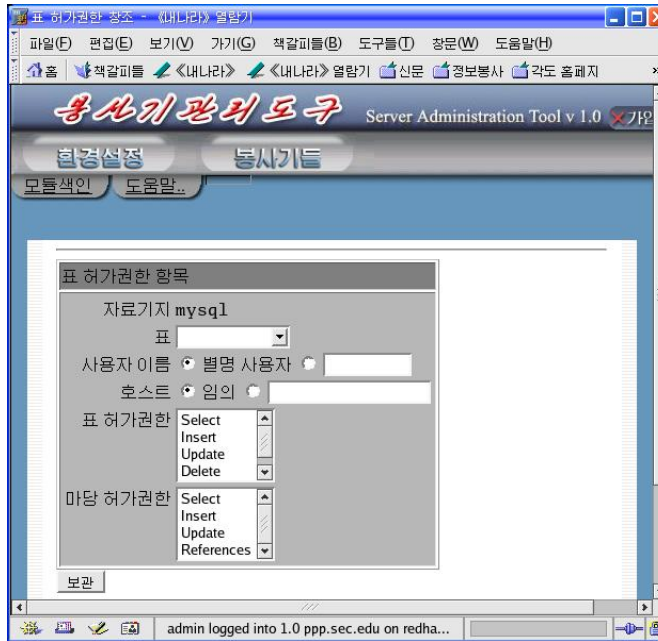



그림 7-35. 봉사기관리도구를 리용한 봉사기관리(6)

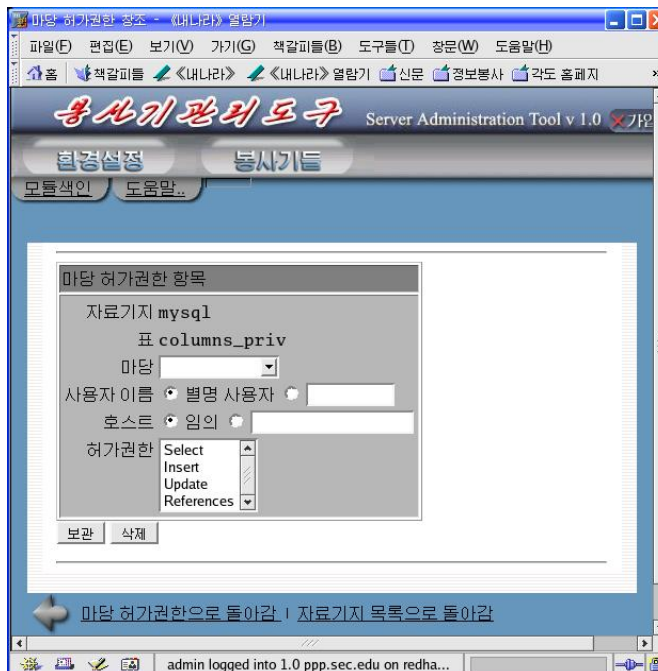



그림 7-36. 봉사기관리도구를 리용한 봉사기관리(7)

봉사기관리도구를 리용한 봉사기의 관리는 앞에서 설명한 내용들을 잘 파악하였으면 쉽게 알수 있으므로 구체적인 설명은 략하고 지나간다.

실지 앞에 있는 그림들만 보고도 어떤 대면부에서는 어떤 관리를 할수 있으며 어떤 설

정항목은 어떻게 처리하여야 하겠는지를 쉽게 알수 있다.

그러면 봉사기관리도구를 리용한 봉사기의 기동과 중지에 대해 보도록 하자.

봉사기의 기동은 봉사기관리도구의 대면부에서《기동》단추를 누르면 된다.(그림 7-37)

봉사기의 설치가 정확히 되고 설정이 제대로 되였으면 아무런 오유통보문이 없이 봉사기가 기동되여 그림 7-38이 나타난다.. 그러나 오유가 있으면 해당한 오유통보문이 현시된다.

봉사기를《중지》시킬 필요가 있으면 그림 7-38과 같은 대면부에서《중지》단추를 누르면 된다.

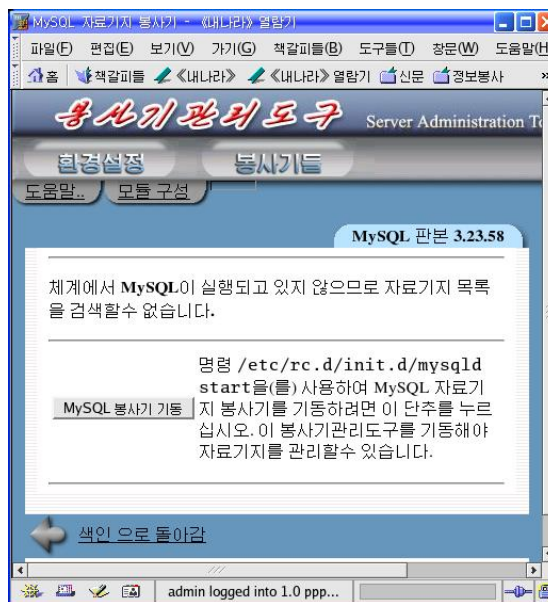



그림 7-37. MySQL봉사기의 기동

그림 7-38. MySQL봉사기의 중지

색 인

Linux봉사기구축기술

집 필	박종혁	심 사	조규철
편 집	김강호	교 정	서금석
장 정	서경애	콤퓨터편성	여은정

낸 곳	교육성 교육정보쎈터	인쇄소	교육성 교육정보쎈터
인 쇄	주체97(2008)년 8월 10일	발 행	주체97(2008)년 8월 20일

교-07-1309